

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

институт

Прикладная информатика, математика и естественнонаучные дисциплины  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

Е. Н. Скуратенко

подпись

инициалы, фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 2017 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.03 – Прикладная информатика

код – наименование направления

Разработка информационной системы аттестационного тестирования  
медицинских работников

тема

Руководитель

\_\_\_\_\_  
подпись, дата

доцент, кпн

должность, ученая степень

И. В. Янченко

инициалы, фамилия

Выпускник

\_\_\_\_\_  
подпись, дата

В. С. Дресвянкин

инициалы, фамилия

Абакан 2017

Продолжение титульного листа БР по теме «Разработка информационной системы аттестационного тестирования медицинских работников»

Консультанты по разделам:

Аналитический  
наименование раздела

\_\_\_\_\_  
подпись, дата

А. В. Ивашина  
инициалы, фамилия

Проектный  
наименование раздела

\_\_\_\_\_  
подпись, дата

А. В. Ивашина  
инициалы, фамилия

Экономический  
наименование раздела

\_\_\_\_\_  
подпись, дата

Е. Н. Скуратенко  
инициалы, фамилия

Нормоконтролер

\_\_\_\_\_  
подпись, дата

В. И. Кокова  
инициалы, фамилия

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

институт

Прикладная информатика, математика и естественнонаучные дисциплины  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

Е. Н. Скуратенко

подпись

инициалы, фамилия

« \_\_\_\_ » \_\_\_\_\_ 2017 г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**в форме бакалаврской работы**  
бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту Дресвянкину Вадиму Сергеевичу  
фамилия, имя, отчество студента

Группа 53-1 (ХБ 13-04) Направление (специальность) 09.03.03  
код

«Прикладная информатика»  
наименование

Тема выпускной квалификационной работы Разработка информационной системы аттестационного тестирования медицинских работников

Утверждена приказом по университету № 156 от 28.02.2017 г.

Руководитель ВКР И. В. Янченко, доцент кафедры «Прикладная информатика, математика и естественнонаучные дисциплины», ХТИ – филиал СФУ  
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР Заказ ГКУЗ РХ «РМИАЦ»

Перечень разделов ВКР:

1. Анализ деятельности Министерства Здравоохранения Республики Хакасия в области аттестационного тестирования.
2. Разработка модуля, реализующего процесс тестирования медицинских работников.
3. Расчет экономического эффекта от внедрения информационной системы аттестационного тестирования медицинских работников.

Перечень графического материала —

Руководитель ВКР \_\_\_\_\_  
подпись

И. В. Янченко  
инициалы, фамилия

Задание принял к исполнению \_\_\_\_\_  
подпись

В. С. Дресвянкин  
инициалы, фамилия

«28» февраля 2017 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка информационной системы аттестационного тестирования медицинских работников» содержит 83 страницы текстового документа, 17 таблиц, 57 изображений, 11 использованных источников и 2 приложения.

АТТЕСТАЦИЯ, СИСТЕМА ТЕСТИРОВАНИЯ, МЕДИЦИНСКИЕ И ФАРМАЦЕВТИЧЕСКИЕ РАБОТНИКИ, БАЗА ДАННЫХ, СУБД MYSQL, PHP, HTML, CSS, JQUERY, МОДУЛЬ ПОЛЬЗОВАТЕЛЯ, ВЕБ-ПРИЛОЖЕНИЕ, ГКУЗ РХ «РМИАЦ».

Объектом исследования данной работы является процесс аттестационного тестирования медицинских и фармацевтических работников.

Предметом – информационная система аттестационного тестирования медицинских и фармацевтических работников.

Цель выпускной квалификационной работы – разработка информационной системы аттестационного тестирования медицинских и фармацевтических работников.

Задачи выпускной квалификационной работы – исследование предметной области; изучение нормативно-правовой базы аттестации медицинских и фармацевтических работников; теоретический анализ средств разработки аттестационного тестирования; выбор средств разработки информационной системы аттестационного тестирования; разработка и создание базы данных для информационной системы аттестационного тестирования; разработка модулей администратора, пользователя (тестируемого), модератора системы аттестационного тестирования.

Разработанная система позволяет осуществлять дистанционное тестирование медицинских и фармацевтических работников Республики Хакасия при прохождении ими аттестации.

## SUMMARY

The theme of the paper is «Development of information system performance testing medical worker» contains 83 pages, 17 tables, 57 images, the list of 11 references and 2 applications.

ATTESTATION, TESTING SYSTEM, MEDICAL AND PHARMACEUTICAL WORKERS, DATABASE, DBMS MYSQL, PHP, HTML, CSS, JQUERY, THE USER MODULE, THE WEB APPLICATION, GKUZ RKH «RMIAC».

The purpose of this paper is to develop of information system attestation testing medical and pharmaceutical workers.

Tasks of this paper are the study of the subject area; review of the regulatory framework of attestation of medical and pharmaceutical workers; theoretical analysis and choice of development tools attestation testing; development and creation of a database for the information system attestation testing; development of modules for the administrator, moderator and users (tested) of the system attestation testing.

The research object of this work is the process of attestation testing of medical and pharmaceutical workers.

A subject is information system attestation testing of medical and pharmaceutical workers.

The developed system allows carrying out remote testing of medical and pharmaceutical workers of the Republic of Khakassia in passing their attestation.

Supervisor in English

\_\_\_\_\_  
signature, date

E. A. Nikitina  
initials, surname

## СОДЕРЖАНИЕ

Введение .....	9
1 Анализ деятельности Министерства Здравоохранения Республики Хакасия в области аттестационного тестирования .....	11
1.1 Общие сведения о Министерстве здравоохранения Республики Хакасия .....	11
1.2 Программные продукты, имеющиеся в распоряжении Министерства здравоохранения Республики Хакасия .....	14
1.3 Аттестация медицинских и фармацевтических работников .....	16
1.4 Анализ ИС тестирования сотрудников .....	24
1.5 Анализ средств разработки информационных систем .....	26
1.5.1 Сравнительный анализ и выбор средств программирования .....	27
1.5.2 Сравнительный анализ и выбор СУБД .....	31
1.5.3 Обзор популярных веб-серверов .....	33
Выводы по аналитическому разделу .....	34
2 Разработка модуля, реализующего процесс тестирования медицинских работников .....	36
2.1 Информационное обеспечение разработки модуля, реализующего процесс тестирования .....	36
2.1.1 Диаграмма потоков данных .....	37
2.1.2 Диаграмма «сущность-связь» .....	39
2.2 Разработка модуля информационной системы, реализующего процесс аттестационного тестирования медицинских работников .....	44
2.2.1 Файловая структура .....	44
2.2.2 Создание функции регистрации пользователя .....	47
2.2.3 Создание функции авторизация пользователя .....	49
2.2.4 Создание личного кабинета пользователя .....	50
2.2.5 Создание функции тестирования .....	52
2.3 Обеспечение информационной безопасности .....	54
2.4 Контрольный пример .....	56
Выводы по проектному разделу .....	71
3 Расчет экономического эффекта от внедрения информационной системы аттестационного тестирования медицинских работников .....	72
3.1 Расчет капитальных затрат .....	72
3.2 Расчет совокупной стоимости владения информационной системой .....	76

3.3 Оценка внедрения ИС как инвестиционного проекта .....	78
Вывод по экономическому разделу .....	79
Заключение.....	81
Список сокращений .....	83
Список использованных источников.....	84
Приложение А Программные продукты.....	86
Приложение Б Программный код системы .....	90



## **ВВЕДЕНИЕ**

В современном обществе возросло внимание общества к компетентности медицинских работников. В связи, с чем появилась необходимость непрерывного совершенствования профессиональных навыков и знаний на протяжении всей жизни, повышения профессионального уровня и квалификации специалистов.

Аттестация медицинских работников на получение категории или подтверждение существующей позволяет подтвердить профессиональные навыки и умения или повысить квалификационную категорию. Состоит из нескольких этапов, одним из которых является тестирование, необходимое для проверки теоретических знаний специалиста. По результатам тестирования решается вопрос о допуске к собеседованию.

Проблема заключается в отсутствии комплексной информационной системы аттестационного тестирования (ИС АТ) на уровне региона.

Объектом исследования данной работы является процесс АТ медицинских работников.

Предметом – информационная система аттестационного тестирования медицинских работников.

Цель выпускной квалификационной работы — разработка информационной системы аттестационного тестирования медицинских работников.

Для достижения поставленной цели определены следующие задачи:

1. исследовать предметную область;
2. изучить нормативно-правовую базу аттестации медицинских и фармацевтических работников;
3. провести теоретический анализ средств разработки аттестационного тестирования;
4. обосновать выбор конкретных средств разработки ИС АТ;
5. разработать и создать БД для ИС АТ;

6. разработать модуль администратора ИС АТ;
7. разработать модуль пользователя системы (тестируемого);
8. разработать модуль модератора системы;
9. протестировать и отладить готовую ИС АТ.

Используемые методы исследования:

- теоретический анализ предметной области для ознакомления с предприятием, выявления его проблем и информационных потребностей; обоснования выбора конкретных средств разработки ИС;
- объектно-ориентированный подход для построения информационно-логической модели процесса аттестационного тестирования;
- сравнение средств разработки информационной системы, для обзора готовых систем тестирования.

Исходя из объемов поставленных задач и временных ограничений, консультантом принято решение разрабатывать систему модульно, разделив ее на две части.

В данной работе рассматривается создание главной страницы сайта, страниц справки, технической поддержки и реализация модуля пользователя, включающая функции их регистрации, авторизации, восстановления забытого пароля, создание личного кабинета пользователей, тестирования, вывода результатов.

Кроме того в экономической части рассмотрены капитальные затраты на создание информационной системы, рассчитана совокупная стоимость владения информационной системой, произведена оценка внедрения информационной системы как инвестиционного проекта.

Разработка модуля администратора и модератора системы рассматривается в работе студента группы 53-1 Султрековой Е. Е.

## **1 Анализ деятельности Министерства Здравоохранения Республики Хакасия в области аттестационного тестирования**

В данном разделе произведен анализ аттестационного тестирования медицинских и фармацевтических работников как одной из областей деятельности Министерства здравоохранения.

Также осуществлены обзор информационных систем тестирования сотрудников, сравнение и выбор средств разработки веб-приложений.

### **1.1 Общие сведения о Министерстве здравоохранения Республики Хакасия**

Министерство здравоохранения Республики Хакасия преобразовано из Хакасского областного отдела здравоохранения на основании Постановления Верховного Совета Республики Хакасия от 10.02.1992 г. № 40-1.

Министерство здравоохранения Республики Хакасия является исполнительным органом государственной власти Республики Хакасия, осуществляющим функции по реализации государственной политики и управлению в сфере здравоохранения, включая вопросы организации оказания медицинской помощи и лекарственного обеспечения, на основании Положения «О Министерстве здравоохранения Республики Хакасия», утвержденном Постановлением Правительства Республики Хакасия от 11 июня 2009 г. № 260.

Полномочия Министерства здравоохранения Республики Хакасия в рамках исследуемой предметной области следующие:

- Обеспечение развитие информационной системы в сфере здравоохранения.
- Создание территориальных аттестационных комиссий для получения квалификационных категорий медицинскими работниками и фармацевтическими работниками.

Министерство возглавляет Министр, назначаемый на должность

(освобождаемый от должности) Главой Республики Хакасия - Председателем Правительства Республики Хакасия.

Министр несет персональную ответственность за выполнение возложенных на Министерство полномочий.

Министр имеет заместителей, назначаемых на должность и освобождаемых от должности министром по согласованию с Главой Республики Хакасия - Председателем Правительства Республики Хакасия.

Количество заместителей министра устанавливает Правительство Республики Хакасия [1].

Структурная схема Министерства Здравоохранения представлена на рисунке 1.

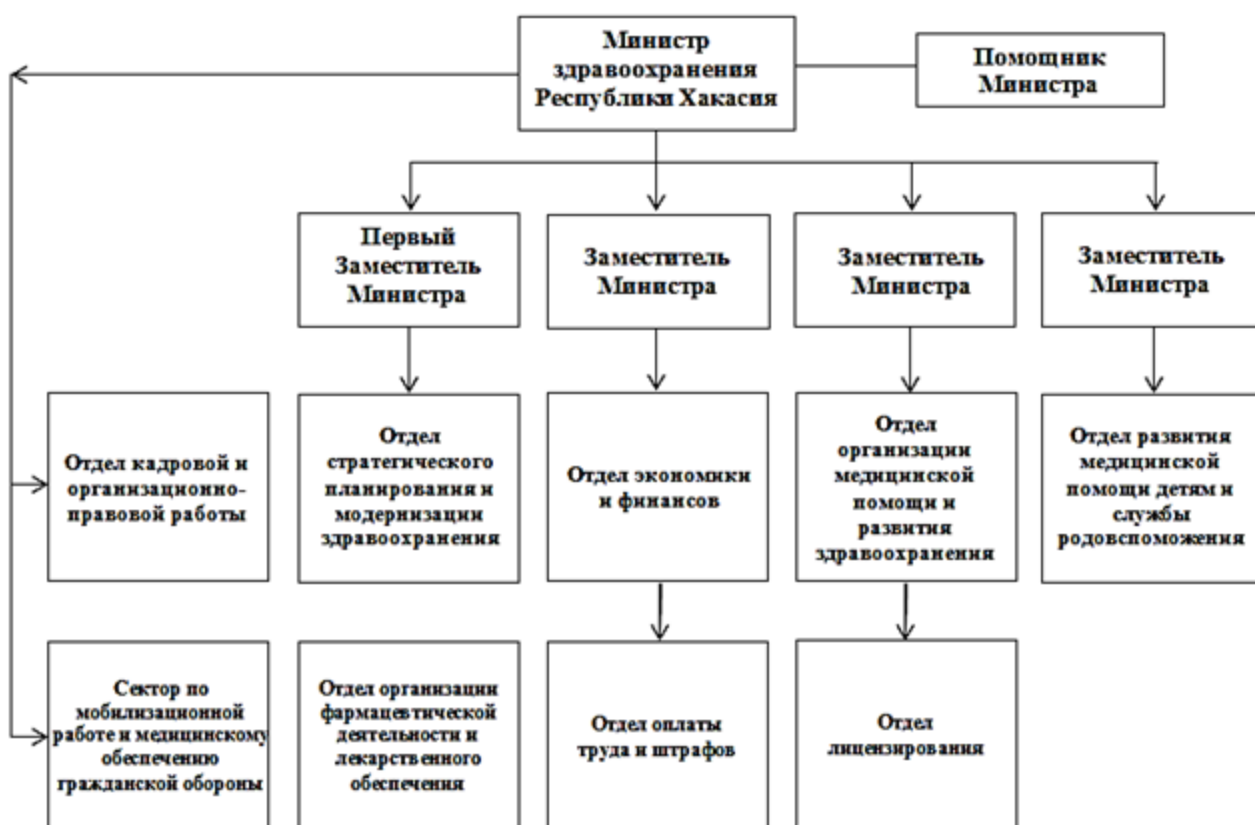


Рисунок 1 – Структурная схема учреждения

Организацией сбора, обработкой и анализом медицинских статистических данных о деятельности учреждений здравоохранения,

состоянии здоровья населения, управлением системой медицинского статистического учета и отчетности в организациях и учреждениях здравоохранения Республики Хакасия занимается ГКУЗ РХ «РМИАЦ».

Государственное казенное учреждение здравоохранения Республики Хакасия «Республиканский медицинский информационно-аналитический центр» является некоммерческой организацией созданной в соответствии с Постановлением Правительства Республики Хакасия от 30.08.2007 № 274. Учреждение находится в ведении Министерства здравоохранения Республики Хакасия.

Основными целями деятельности учреждения в рамках исследуемой предметной области являются:

1. Организационно-методическое руководство по формированию единой информационной системы здравоохранения в Республике Хакасия, созданию и сопровождению автоматизированных систем управления здравоохранением Республики Хакасия.

2. Повышение эффективности использования информационной структуры здравоохранения Республики Хакасия, обеспечение достоверности сведений в учетной и отчетной медицинской документации.

3. Внедрение новых технологий сбора и обработки медико-статистической информации.

4. Обеспечение безопасности автоматизированных информационных систем здравоохранения Республики Хакасия.

5. Обеспечение информационной поддержки по реализации приоритетного национального проекта «Здоровье».

6. Формирование единой информационной системы здравоохранения Российской Федерации на базе современных компьютерных технологий межотраслевой системы сбора, обработки, хранения и предоставления информации, обеспечивающей динамическую оценку состояния здоровья и информационную поддержку принятия решений, направленных на его улучшение [2].

## **1.2 Программные продукты, имеющиеся в распоряжении Министерства здравоохранения Республики Хакасия**

Под IT-инфраструктурой организации понимается вся совокупность имеющихся в ней сервисов и систем, сетей, технических и программных средств, данных, автоматизированных процессов.

Информационные ресурсы являются основой для создания информационных продуктов. Любой информационный продукт отражает информационную модель его производителя и воплощает его собственное представление о некотором объекте конкретной предметной области. Информационный продукт, являясь результатом интеллектуальной деятельности человека, должен быть зафиксирован на материальном носителе в виде документов, статей, обзоров, программ, книг и т. д.

Создание и сопровождение автоматизированных систем управления здравоохранением Республики Хакасия – одна из функций Республиканского медицинского информационно-аналитического центра.

Электронные медицинские карты ведутся в 29 медицинских организациях. Доля пациентов, у которых ведутся электронные медицинские карты, составляет 61% от населения Республики Хакасия.

Предоставление государственной услуги «Заявка (запись) на прием к врачу» в электронном виде осуществляется с помощью электронной регистратуры, веб-сервис [kvrachu19.rf](http://kvrachu19.rf).

Сведения о медицинских работниках Республики Хакасия хранятся в Федеральном регистре медицинских работников, а о медицинских организациях и их структурных подразделениях, оказывающих медицинскую помощь – в Федеральном регистре медицинских организаций.

В программном комплексе АСУЛОН осуществляется выписка рецептов на лекарственные средства и изделия медицинского назначения.

Также медицинские организации Министерства Здравоохранения используют систему «Автограф» для мониторинга санитарного,

автомобильного транспорта, позволяющая видеть точное местоположение машины на карте и измерять простой и пробег, выбирать целесообразный маршрут движения.

Полный перечень программных продуктов, имеющихся в распоряжении органа управления здравоохранением, представлен в приложении А.

Тестовый контроль медицинских работников осуществляется дистанционно. На сайте Министерства здравоохранения Республики Хакасия расположена ссылка на сайт системы тестирования, расположенной по адресу <http://tms.miacrh.ru/>.

Действующая система разработана специалистами ГКУЗ РХ «РМИАЦ» в 2012 году. Общий вид приложения представлен на рисунке 2.

**TMC**  
Система тестирования медицинских работников 2.1

✓ TMC Главная Регистрация Техническая поддержка Справка

**Авторизация**

Логин

Пароль

Вход

Регистрация  
для прохождения тестирования.

График медицинского и фармацевтического персонал

График среднего медицинского персонала

Copyright © 2012-2014 Медицинский информационно-аналитический центр Республики Хакасии

Телефон технической поддержки: 8 (3902) 295-025.  
Режим работы: 8-00 до 16-30.

Рисунок 2 – Главная страница сайта системы тестирования

Основными недостатками системы являются отсутствие личного кабинета пользователей, отсутствие проверки пользователя (является ли он медицинским работником).

Личный кабинет необходим для предоставления пользователю возможностей изменения личной информации в случае необходимости, а также просмотра всех результатов тестирований. Это необходимо для сокращения числа звонков пользователей в техническую поддержку.

Проверка медицинских работников необходима для пресечения доступа к

тестовым вопросам посторонних лиц. На данный момент не обязательно быть медицинским работником, достаточно иметь почтовый ящик, чтобы зарегистрироваться в системе и пройти тестирование.

В связи с вышесказанным, появилась необходимость модернизации текущей системы или замена ее более совершенной.

### **1.3 Аттестация медицинских и фармацевтических работников**

Аттестация медицинских и фармацевтических работников на категорию реализует их право подтвердить свои профессиональные навыки и умения или повысить квалификационную категорию.

Данное право предоставлено медицинскому работнику согласно приказа Министерства Здравоохранения Российской Федерации от 23 апреля 2013 г. №240н. При этом аттестация является добровольной и проводится аттестационными комиссиями по трем квалификационным категориям: второй, первой и высшей.

Требования к специалистам, претендующим на получение каждой квалификационной категории, представлены на рисунках 3, 4 и 5 соответственно.

<b>Вторая категория</b>	<ul style="list-style-type: none"><li>- иметь теоретическую подготовку и практические навыки в области осуществляемой профессиональной деятельности;</li><li>- использовать современные методы диагностики, профилактики, лечения, реабилитации и владеть лечебно-диагностической техникой в области осуществляемой профессиональной деятельности;</li><li>- ориентироваться в современной научно-технической информации, владеть навыками анализа количественных и качественных показателей работы, составления отчета о работе;</li><li>- иметь стаж работы по специальности (в должности) не менее трех лет.</li></ul>
-------------------------	---

Рисунок 3 – Требования к специалистам второй категории



<b>Первая категория</b>	<ul style="list-style-type: none"> <li>- иметь теоретическую подготовку и практические навыки в области осуществляемой профессиональной деятельности и смежных дисциплин;</li> <li>- использовать современные методы диагностики, профилактики, лечения, реабилитации и владеть лечебно-диагностической техникой в области осуществляемой профессиональной деятельности;</li> <li>- уметь квалифицированно провести анализ показателей профессиональной деятельности и ориентироваться в современной научно-технической информации;</li> <li>- участвовать в решении тактических вопросов организации профессиональной деятельности;</li> <li>- иметь стаж работы по специальности (в должности) не менее пяти лет.</li> </ul>
-------------------------	--

Рисунок 4 – Требования к специалистам первой категории

<b>Высшая категория</b>	<ul style="list-style-type: none"> <li>- иметь высокую теоретическую подготовку и практические навыки в области осуществляемой профессиональной деятельности, знать смежные дисциплины;</li> <li>- использовать современные методы диагностики, профилактики, лечения, реабилитации и владеть лечебно-диагностической техникой в области осуществляемой профессиональной деятельности;</li> <li>- уметь квалифицированно оценить данные специальных методов исследования с целью установления диагноза;</li> <li>- ориентироваться в современной научно-технической информации и использовать ее для решения тактических и стратегических вопросов профессиональной деятельности;</li> <li>- иметь стаж работы по специальности (в должности) не менее семи лет.</li> </ul>
-------------------------	---

Рисунок 5 – Требования к специалистам высшей категории

Аттестация специалистов со средним и высшим медицинским и фармацевтическим образованием проводится один раз в пять лет. По истечении этого срока производится повышение категории или ее подтверждение.

При аттестации оцениваются теоретические знания и практические навыки, необходимые для выполнения профессиональных обязанностей по соответствующим специальностям и должностям, на основе результатов квалификационного экзамена.

Квалификационный экзамен включает в себя экспертную оценку отчета о

профессиональной деятельности специалиста, тестовый контроль знаний и собеседование.

Тестовый контроль предусматривает выполнение специалистом тестовых заданий и признается пройденным при условии успешного выполнения не менее 70% общего объема тестовых заданий.

Аттестационная комиссия состоит из Координационного комитета, осуществляющего функции по организации деятельности аттестационной комиссии, и экспертных групп по специальностям, осуществляющих аттестацию специалистов в части рассмотрения документов и проведения квалификационного экзамена.

В состав аттестационной комиссии включаются ведущие специалисты организаций, осуществляющих медицинскую и фармацевтическую деятельность [3].

Функциональная модель процессов АТ медицинских работников будет спроектирована в нотации IDEF0.

IDEF0 используется для создания функциональной модели, которая является структурированным изображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции.

Методология IDEF0 может использоваться для моделирования широкого круга систем, где под системой понимается любая комбинация средств аппаратного и программного обеспечения, а также людей. При создании новых систем IDEF0 может вначале применяться для определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. Для уже существующих систем, IDEF0 может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.

Результатом применения методологии IDEF0 является модель. Модель состоит из диаграмм, фрагментов текста и глоссария, которые имеют ссылки друг на друга. Диаграммы - главные компоненты модели. На диаграммах все

функции производственной системы и интерфейсы представлены как блоки (функции) и дуги (интерфейсы). Управляющие производством данные входят в блок сверху, в то время как материалы или информация, которые подвергаются производственной операции, показаны с левой стороны блока; результаты выхода показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рис. 6) [4].

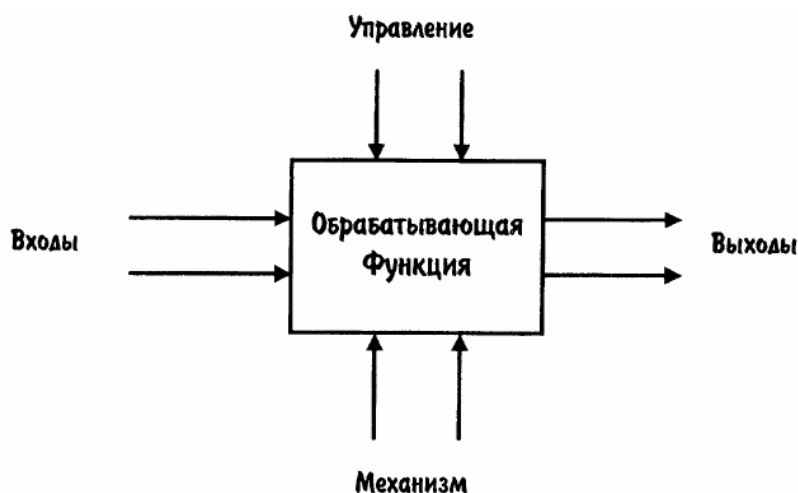


Рисунок 6 – Функциональный блок в IDEF0-модели

Диаграмма потоков данных аттестации медработников по модели «Черный ящик» представлена на рисунке 7.

Дугами механизма для процесса аттестации будут являться Информационная система аттестационного тестирования и Аттестационная комиссия, состоящая из Координационного комитета и Экспертной группы.

Дуга управления – приказ Министерства здравоохранения Российской Федерации от 23 апреля 2013 г. N 240н «О порядке и сроках прохождения медицинскими работниками и фармацевтическими работниками аттестации для получения квалификационной категории».

Одной из наиболее важных особенностей методологии IDEF0 является постепенное введение все больших уровней детализации по мере создания

диаграмм, отображающих модель. Декомпозиция диаграммы первого уровня представлена на рисунке 8.

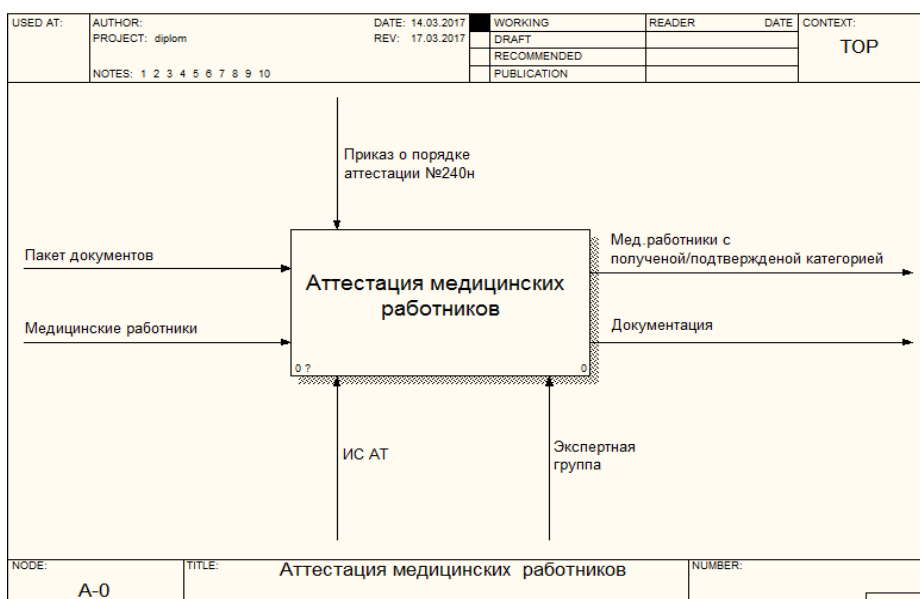


Рисунок 7 – Диаграмма потоков данных

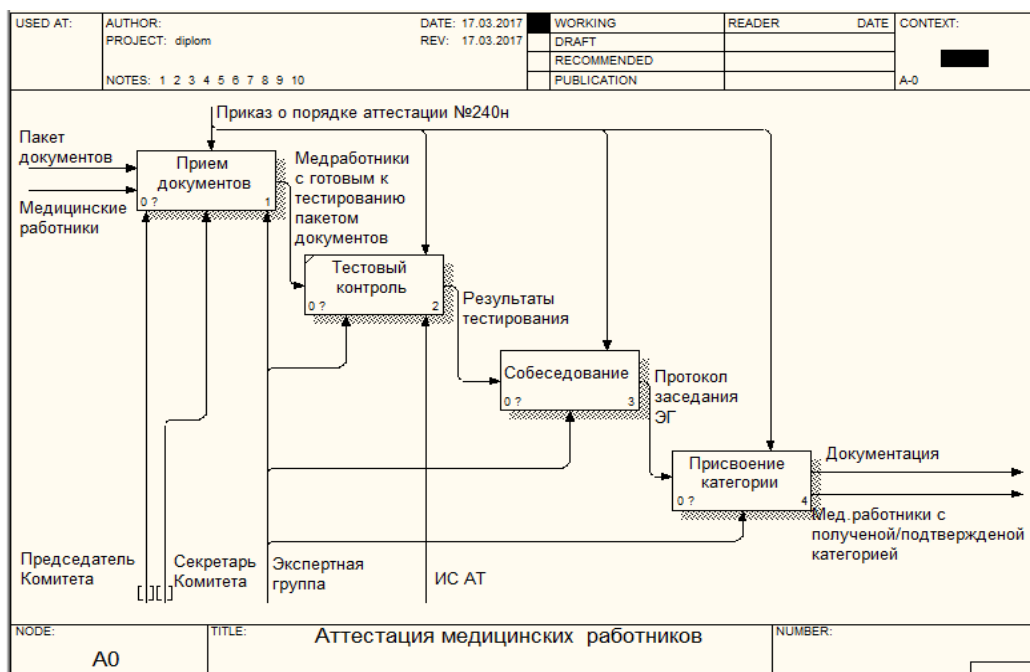


Рисунок 8 – Декомпозиция первого уровня диаграммы потоков данных

Аттестация медицинских работников начинается с приема документов секретарем комитета, их регистрации, проверки наличия предусмотренных

Приказом документов и правильности их оформления.

В случае наличия всех документов и при условии правильности их заполнения медицинский работник проходит тестовый контроль. Исходя из результатов которого, решается вопрос о допуске сотрудника к собеседованию.

При условии успешного выполнения не менее 70% тестовых заданий, медицинский работник допускается к собеседованию по теоретическим и практическим вопросам профессиональной деятельности с членами Экспертной группы.

По результатам собеседования принимается решение о присвоении или отказе в присвоении квалификационной категории, которое оформляется протоколом заседания Экспертной группы.

Более детальное описание каждого процесса представлено на диаграммах декомпозиций второго уровня.

Декомпозиция процесса «Прием документов» (рис. 9).

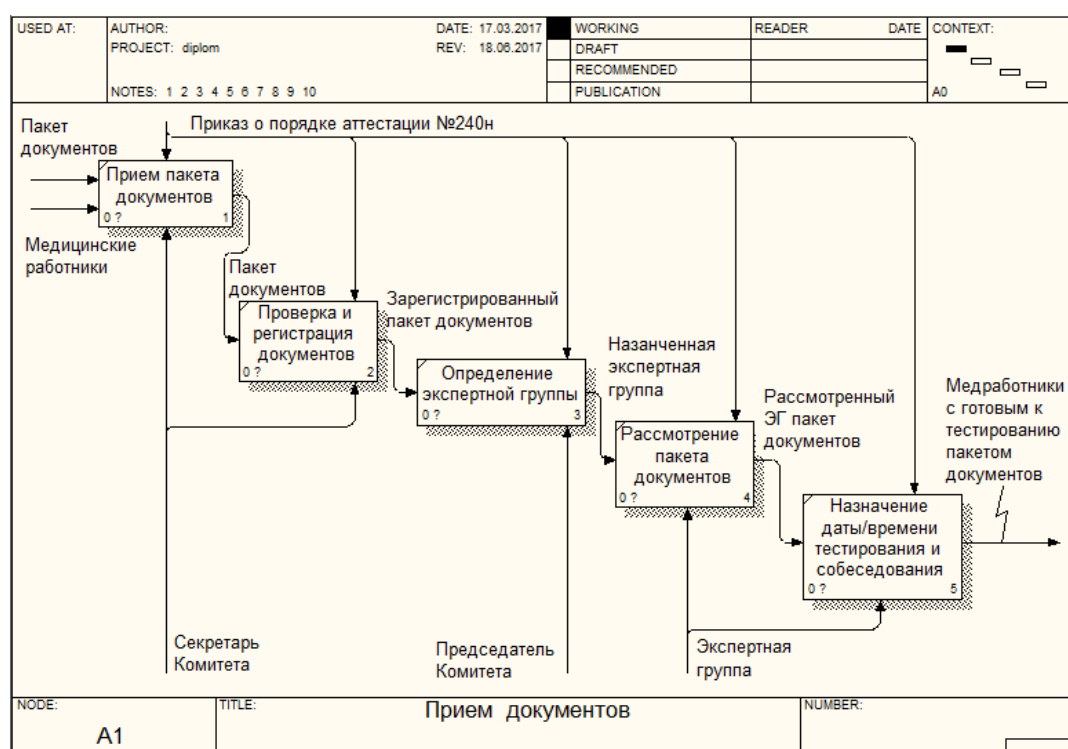


Рисунок 9 – Декомпозиция второго уровня функции «Прием документов»

Помимо описанных выше приема и проверки пакета документов председателем комитета производится определение экспертной группы, которая будет производить аттестацию. Рассмотрение принятого пакета документов и назначение даты и времени проведения тестирования и собеседования назначенной экспертной группой.

Процесс собеседования состоит из двух функций: проведение самого собеседования по теоретическим и практическим вопросам профессиональной деятельности специалиста и принятие решения экспертной группой (рис. 10). После этого оформляется протокол заседания Экспертной группы, в который заносится результат собеседования и отправляется в Комитет.

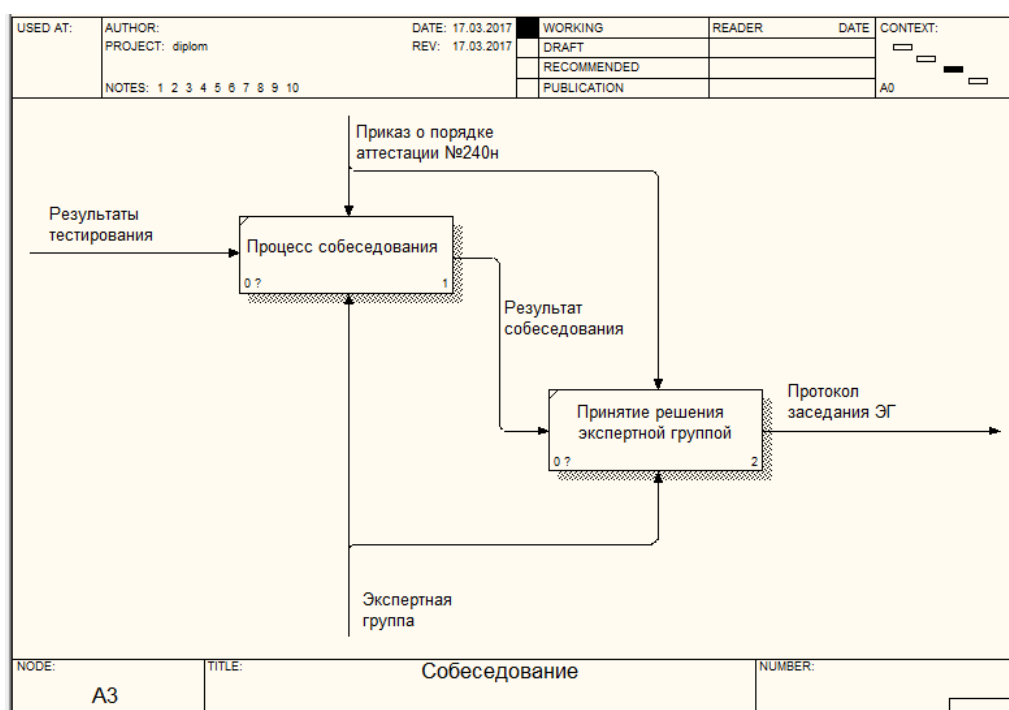


Рисунок 10 – Декомпозиция второго уровня функции «Собеседование»

Декомпозиция функции «Присвоение категории» представлено на рисунке 11.

Процесс утверждения принятого экспертной группой решения осуществляет председатель Комитета. Результатом утверждения является подготовленный распорядительный акт о присвоении квалификационной

категории специалисту. Данный акт для утверждения отправляется в орган государственной власти или организацию, создавшую аттестационную комиссию.

После утверждения акта производится выдача выписки из распорядительного акта о присвоении специалисту квалификационной категории, которая регистрируется в журнале регистрации документов.

Для того чтобы быть правильно понятыми, существуют словари описания активностей и стрелок. Словарь стрелок диаграммы IDEF0 представлен в таблице 1.

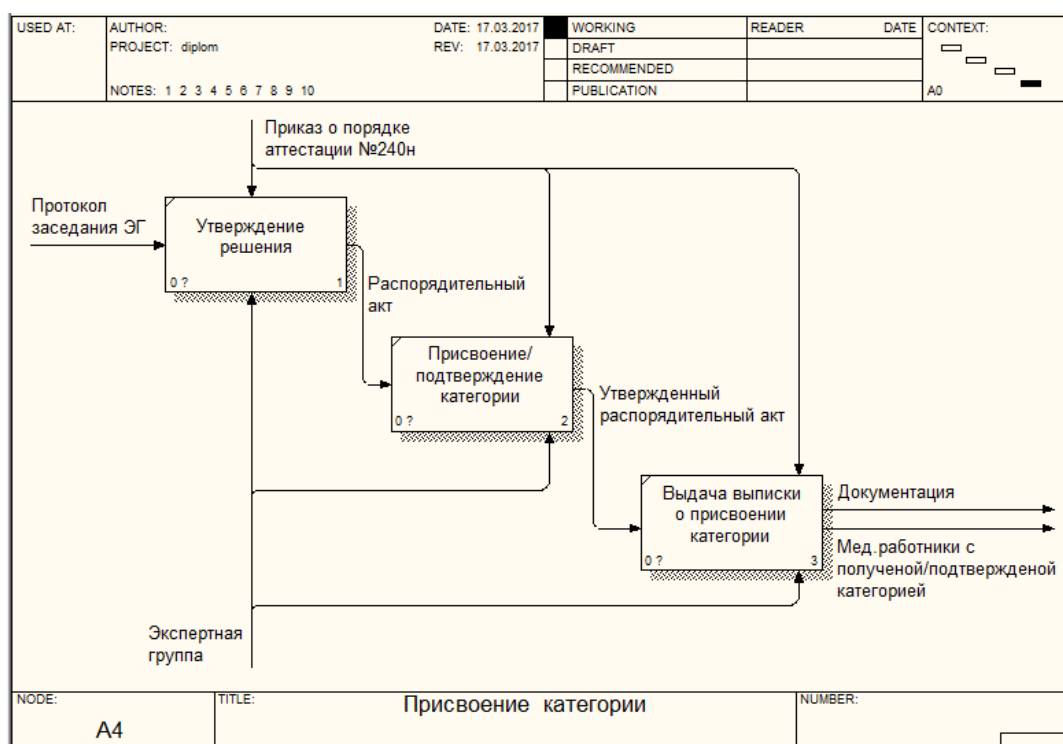


Рисунок 11 – Декомпозиция второго уровня функции «Присвоение категории»

Таблица 1 – Таблица словарей стрелок

Наименование стрелки	Пояснение
Пакет документов (вход)	Перечень документов, подаваемых специалистами в аттестационную комиссию, определенный приказом №240н
Медицинские работники (вход)	Сотрудники, имеющие медицинское и фармацевтическое образование, изъявившие желание пройти аттестацию для получения квалификационной категории

## Окончание таблицы 1

Приказ о порядке аттестации №240н (управление)	Приказ Министерства здравоохранения РФ от 23 апреля 2013 г. «О порядке и сроках прохождения медицинскими работниками и фармацевтическими работниками аттестации для получения квалификационной категории»
ИС АТ (механизм)	Информационная система аттестационного тестирования медицинских работников, позволяющая автоматизировать процесс тестового контроля знаний
Аттестационная комиссия (механизм)	Команда специалистов организаций, осуществляющих медицинскую и фармацевтическую деятельность, созданная для проведения аттестации. Аттестационная комиссия состоит из Координационного комитета и Экспертных групп по специальностям.
Мед. работники с полученной/подтвержденной категорией (выход)	Сотрудники, имеющие медицинское и фармацевтическое образование, успешно прошедшие все этапы аттестации и получившие/подтвердившие квалификационную категорию
Документация (выход)	Набор документов, скопившихся в процессе аттестации и подлежащих дальнейшему хранению.

### 1.4 Анализ ИС тестирования сотрудников

Компьютерное тестирование знаний – один из вариантов реализации тестирования, традиционно рассматриваемого как форма измерения знаний, основанная на применении тестов.

Для сравнения выбраны четыре популярные системы тестирования: UniTest, INDIGO, Moodle и testserver.pro.

UniTest – это комплексное программное решение для проведения компьютерного тестирования.

Программа предназначена для формирования банка тестовых заданий и организации процесса проверки знаний. Позволяет организовать процесс контроля знаний с помощью компьютерного тестирования в сети с использованием транспортного протокола TCP/IP, а также локального тестирования.

Indigo – представляет собой multifunctional комплекс



программного обеспечения, позволяющий автоматизировать процесс проведения тестирования и обработки результатов.

Moodle – это система управления содержимым сайта, специально разработанная для создания онлайн-курсов преподавателями. Система тестирования является лишь малой частью большой программы. Moodle – это абсолютно бесплатный проект с открытым исходным кодом.

Система тестирования testserver.pro позволяет проводить аттестацию сотрудников, как по своим, так и по чужим Бадам Знаний [5].

Сравнение систем тестирования представлено в таблице 2.

Таблица 2 – Сравнение систем тестирования

	UniTest	INDIGO	Moodle	testserver.pro
Модуль администратора	+	+	+	+
Модуль пользователя	+	+	+	+
Самостоятельная регистрация тестируемых	-	+	По умолчанию – ручная	-
Дополнительные виды тестовых вопросов, кроме основных	+	-	-	-
Возможность импорта вопросов	-	+	+	-
Возможность экспорта таблиц с результатами	-	+	+	-
Особенности технической реализации	Как клиент-серверное приложение, так и локально	Модуль администратора – локальное приложение, модуль пользователя веб-приложение	Клиент-серверное приложение	Клиент-серверное приложение
Стоимость	Бесплатно	10 000 рублей за 10 соединений + облако 20 000 рублей в год + 50% от стоимости лицензии в год за техническое обслуживание со второго года	Бесплатно	Бесплатно

Система UniTest не может быть использована, так как имеет ряд существенных недостатков: несовместима с последними версиями ОС (Windows 7/8); назначение логина и пароля, для аутентификации преподавателей и студентов, непосредственно «Администратором» («в ручную»).

Система «INDIGO» также не будет выбрана, так как лицензия является платной.

Использование Moodle нецелесообразно, так как система тестирования является частью большого программного продукта; обслуживание предоставляется за отдельную плату.

Также отказываемся от системы тестирования testserver.pro, так как регистрация пользователей осуществляется в ручную, отсутствуют возможности импорта вопросов из уже имеющегося банка и экспорта отчетности.

Исходя из всего вышесказанного, принято решение собственной разработки информационной системы аттестационного тестирования, удовлетворяющей всем требованиям заказчика.

## **1.5 Анализ средств разработки информационных систем**

Под информационной системой обычно понимается прикладная программная подсистема, ориентированная на сбор, хранение, поиск и обработку информации. Подавляющее большинство информационных систем работает в режиме диалога с пользователем.

По способу организации информационные системы подразделяются на следующие классы:

- системы на основе архитектуры файл-сервер;
- системы на основе архитектуры клиент-сервер;
- системы на основе многоуровневой архитектуры;
- системы на основе Интернет/интранет-технологий.

В архитектуре *файл-сервер* сетевое разделение компонентов диалога отсутствует, а компьютер используется для функций отображения, что облегчает построение графического интерфейса. Файл-сервер только извлекает данные из файлов, так что дополнительные пользователи и приложения лишь незначительно увеличивают нагрузку на центральный процессор. Каждый новый клиент добавляет вычислительную мощность к сети.

Архитектура *клиент-сервер* предназначена для разрешения проблем файл-серверных приложений путем разделения компонентов приложения и размещения их там, где они будут функционировать наиболее эффективно. Особенностью архитектуры клиент-сервер является наличие выделенных серверов баз данных, понимающих запросы на языке структурированных запросов (Structured Query Language, SQL) и выполняющих поиск, сортировку и агрегирование информации.

Объектами разработки в таких приложениях помимо диалога и логики обработки являются, прежде всего, реляционная модель данных и связанный с ней набор SQL-операторов для типовых запросов к базе данных.

Многоуровневая архитектура состоит из трех уровней: приложения клиентов, выделенные для выполнения функций и логики представлений; сервер приложений, на котором выполняется прикладная логика и с которого логика обработки данных выполняет операции с базой данных; удаленный специализированный сервер базы данных, выделенный для услуг обработки данных и файловых операций (без использования хранимых процедур).

В развитии Интернет/Интранет-технологий основной акцент пока что делается на разработке инструментальных программных средств. В то же время наблюдается отсутствие развитых средств разработки приложений, работающих с базами данных [6].

### **1.5.1 Сравнительный анализ и выбор средств программирования**

Веб-приложения представляют собой особый тип программ, построенных

по архитектуре «клиент-сервер». Особенность заключается в том, что само веб-приложение находится и выполняется на сервере, клиент при этом получает только результаты работы. Работа приложения основывается на получении запросов от пользователя (клиента), их обработке и выдачи результата.

К наиболее типичным операциям, выполняемым веб-приложениями, относятся:

- прием данных от пользователя и сохранение их на сервере;
- выполнение различных действий по запросу пользователя: извлечение данных из базы данных (БД), добавление, удаление, изменение данных в БД;
- аутентификация пользователя и отображение интерфейса системы, соответствующий данному пользователю;
- отображение оперативной постоянно изменяющейся информации.

Для создания веб-приложений используются различные технологии и языки программирования. Языки веб-программирования делятся на клиентские и серверные.

Клиентские языки обрабатываются на стороне пользователя (браузером). Серверные языки выполняются непосредственно на самом сервере, и пользователь получает только результат работы программы.

По результатам исследования, проводившегося с августа 2014 по апрель 2016 года, сформированного на основе анкетирования 445 digital-компаний, наиболее популярными серверными языками можно назвать PHP, Perl, Python, Ruby.

Самым распространённым клиентским языком веб-разработки является JavaScript.

Также к разработке клиентской части веб-приложения можно отнести технологии HTML и CSS. HTML – язык гипертекстовой разметки, определяет как будет отображаться страница сайта. CSS – каскадные таблицы стилей, служит для оформления и форматирования отображения содержимого.

Ruby – динамический императивный объектно-ориентированный язык программирования. Характеризуется динамической типизацией и

автоматическим управлением памятью. Язык Ruby используется в веб-разработке в составе открытого веб-фреймворка Rails, чаще называемого Ruby on Rails (RoR).

Python широко применяется как интерпретируемый язык для скриптов различного назначения. Python – мультипарадигмальный язык программирования: он позволяет совмещать процедурный подход к написанию кода с объектно-ориентированным и функциональным

PHP – язык программирования, исполняемый на стороне веб-сервера, спроектированный в качестве инструмента создания динамических и интерактивных веб-сайтов [7].

Сравнение языков будет производиться по следующим признакам:

1. Типизация:

- Динамическая типизация. Переменные и параметры методов/функций связываются с типами в момент присваивания значения. Одна и та же переменная в разные моменты может хранить значения разных типов.

- Неявная типизация. Типы переменных и параметров не указываются явно. Неявная типизация может быть и статической, в таком случае типы переменных и параметров вычисляются компилятором.

2. Управление потоком вычислений:

- Поддержка инструкции goto (безусловный переход на метку).
- Поддержка инструкции break без метки (безусловный выход из ближайшего цикла), и соответствующей инструкции continue.
- Поддержка блока else при обработке исключений.

3. Типы и структуры данных:

- Наличие встроенных в язык многомерных массивов.
- Наличие встроенных в язык динамических массивов (способных изменять свой размер во время выполнения программы).
- Наличие встроенных в язык ассоциативных массивов или хэш-таблиц.
- Наличие возможности перебрать все элементы коллекции с помощью цикла foreach.

- Наличие списковых включений (или их аналога).
- Поддержка целых чисел неограниченной разрядности. Должна быть возможность записать сколь угодно большое целое число.
- Возможность определить тип, значениями которого могут быть целые числа только определенного интервала, например [-5..27], при этом присвоение переменной такого типа значения, выходящего за указанные рамки, должно вызывать ошибку.

#### 4. Объектно-ориентированные возможности:

- Интерфейсы. Семантическая и синтаксическая конструкция в коде программы, используемая для специфицирования услуг, предоставляемых классом.
- Возможность в наследнике изменить имя поля/метода предка.
- Множественное наследование. Возможность наследовать класс сразу от нескольких классов.
- Возможность создавать переменные, имеющие динамическую область видимости.
- Именованные параметры. Возможность при вызове функции/метода указывать имена параметров и менять их местами.
- Возможность при вызове функции/метода опускать некоторые параметры, чтобы при этом подставлялось значение по умолчанию, указанное при определении функции.
- Локальные функции. Возможность определять локальную функцию внутри другой функции/метода. Подразумевается возможность использовать внутри локальной функции локальные переменные из внешнего блока.

Результат сравнения представлен в таблице 3, где «+» указанная возможность присутствует; «-» указанная возможность отсутствует; «+/-» возможность поддерживается не полностью; «?» нет данных.

По результатам сравнения язык РНР имеет значительное преимущество. Помимо этого можно выделить еще несколько достоинств:

- развитая поддержка баз данных;

- наличие огромного количества библиотек и расширений языка;
- возможность использования в изолированной среде.

Таблица 3 – Сравнения языков веб-программирования

Критерий	Perl	PHP	Python	Ruby
Динамическая типизация	+	+	+	–
Неявная типизация	+	+	+	+
Инструкция goto	+/-	–	+/-	–
Инструкции break без метки	+	+	+	–
Блок else (исключения)	–	+	+	–
Многомерные массивы	+/-	+/-	+/-	+/-
Динамические массивы	+/-	+/-	+/-	+
Ассоциативные массивы	+	+	+	+
Цикл foreach	+	+	+	+
Списковые включения	–	+	?	+
Целые числа произвольной длины	+/-	+	+	+
Целые числа с контролем границ	–	–	?	–
Интерфейсы	+	+	?	?
Переименование членов при наследовании	–	–	?	?
Множественное наследование	–	+	–	?
Динамические переменные	–	–	?	?
Именованные параметры	–	+	+	+
Значения параметров по умолчанию	+	+	+	+
Локальные функции	+	+	+	+

### 1.5.2 Сравнительный анализ и выбор СУБД

Системы управления базами данных (СУБД) позволяют структурировать, систематизировать и организовывать данные для их компьютерного хранения и обработки. Именно системы управления базами данных являются основой практически любой информационной системы.

Взаимодействие пользователя с БД обеспечивает специальный язык запросов - Structured Query Language (SQL) (Язык структурированных запросов).

По способу доступа к базе данных СУБД подразделяются на файл-серверные, клиент-серверные и встраиваемые.

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. СУБД располагается на каждом клиентском компьютере. Доступ СУБД к данным осуществляется через локальную сеть.

Клиент-серверная СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме. Все клиентские запросы на обработку данных обрабатываются клиент-серверной СУБД централизованно.

Встраиваемая СУБД – это библиотека, которая позволяет унифицированным образом хранить большие объемы данных на локальных машинах.

Существует множество СУБД для управления реляционными базами данных. Каждый из них обладает своими достоинствами и недостатками. Рассмотрим некоторые из них.

Таблица 4 – Сравнение СУБД

	MySQL	PostgreSQL	Oracle	SQL Server
Интерфейс	SQL	SQL	GUI, SQL	GUI, SQL
Поддержка языков	C, C++, Java, Ruby, PHP и др.	Python, Ruby, PHP	C, C++, Java, Ruby, и др.	Java, Ruby, Python, VB, .Net, PHP
Кроссплатформенность (Windows/Linux)	++	++	++	+-
Лицензия	Свободная	Свободная	Проприетарная	Проприетарная
Ответ базы данных	+	+	+	+
Различные типы таблиц	+	-	+	-
Контроль доступа к данным	+	+	+	+
Импорт и экспорт таблиц баз данных	+	+	+	+
Контроль целостности базы данных	+	+	+	+

Использование СУБД Oracle и MS SQL Server невозможно из-за высоких стоимостей сервера и поддержки.

Для веб-приложений главное это производительность и скорость, поэтому MySQL будет лучшим выбором. Дополнительными преимуществами



будут являться:

- многопоточность, поддержка нескольких одновременных запросов;
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- гибкая система привилегий и паролей;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- интерфейс с языками C и Perl, PHP;
- быстрая работа, масштабируемость;
- совместимость с ANSI SQL;
- наличие визуального веб-интерфейса.

### **1.5.3 Обзор популярных веб-серверов**

Веб-сервер — это сетевое приложение, обслуживающее HTTP-запросы от клиентов, обычно веб-браузеров. Веб-сервер принимает запросы и возвращает ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

Созданием программного обеспечения веб-серверов занимаются многие разработчики, но наибольшую популярность по статистике Netcraft имеют такие программные продукты, как Apache (Apache Software Foundation), IIS (Microsoft), Google Web Server и nginx.

Apache — свободное программное обеспечение. Apache уже многие годы является лидером по распространенности во Всемирной паутине в силу своей надежности, гибкости, масштабируемости и безопасности.

IIS (Internet Information Services) — проприетарный набор серверов для нескольких служб Интернета, разработанный Microsoft и распространяемый с серверными операционными системами семейства Windows. Основным компонентом IIS является веб-сервер, также поддерживаются протоколы FTP, POP3, SMTP, NNTP.

Google Web Server — разработка компании Google на основе веб-сервера Apache. GWS оптимизирован для выполнения приложений сервиса Google Applications.

nginx — это HTTP-сервер, совмещенный с кэширующим прокси-сервером. Осенью 2004 года вышел первый публично доступный релиз, сейчас nginx используется на 9-12% веб-серверов [8].

По результатам исследования популярности http-серверов компании Netcraft, построенного на основе автоматизированной проверки более 876 миллионов хостингов, которое было опубликовано 19.01.2015. Apache занимает 50.72% (+0.15) доли рынка (89.8 млн хостингов), nginx — 14.82% (+0.09), IIS — 10.55% (-1.17) и Google Web Server — 8.12%.

### **Выводы по аналитическому разделу**

В данном разделе проведен анализ деятельности МЗ РХ в области аттестационного тестирования, изучена нормативно-правовая база аттестации медицинских и фармацевтических работников, построена функциональная модель процессов АТ.

В результате выявлена необходимость модернизации текущей системы или полной ее замены.

После анализа рынка систем тестирования принято решение о собственной разработке системы. Данное решение обосновано недостаточной функциональностью данных систем, необходимостью доработки их под конкретные требования заказчика, их высокой стоимостью.

Для решения поставленной задачи проведен анализ и выбор средств разработки веб-приложений.

Основным языком веб-разработки будет являться PHP. Данный выбор обусловлен высокой популярностью языка, развитой поддержкой баз данных, кроссплатформенностью языка, высокой скоростью обработки сценариев,

гибкими и эффективными средствами безопасности, бесплатным распространением.

Также будут использованы HTML, CSS, JavaScript.

В качестве системы управления реляционными базами данных выбрано средство MySQL с веб-интерфейсом phpMyAdmin. Выбор сделан исходя из широкой распространенности СУБД MySQL, высокой скорости работы, производительности, надежности и полным взаимодействием с PHP. А визуальный интерфейс phpMyAdmin значительно упрощает работу с базами данных.

Для работы системы будет использоваться веб-сервер Apache в виду его надежности, гибкости, безопасности и масштабируемости.

## **2 Разработка модуля, реализующего процесс тестирования медицинских работников**

Функциональная часть системы написана на скриптовом языке общего назначения для разработки веб-приложений – PHP (версии 7.1.3).

Общие характеристики используемых технологий и инструментов данной платформы:

- сервер баз данных/СУБД: MySQL (версии 5.7.17-log - MySQL Community Server (GPL));
- веб-сервер: Apache/2.4.25;
- веб-приложение для администрирования СУБД MySQL: PhpMyAdmin 4.6.6;
- текстовый редактор notepad++; Sublime Text 3.

Система представляет собой веб-сайт с интуитивно понятным интерфейсом, запущенный в процессе разработки и тестирования на локальном сервере.

Система оперирует информацией из базы данных MySQL с кодировкой таблиц utf8\_general\_ci.

### **2.1 Информационное обеспечение разработки модуля, реализующего процесс тестирования**

Информационное обеспечение ИС включает два блока:

- внешнее информационное обеспечение (классификаторы технико-экономической информации, документы, методические инструктивные материалы);
- внутримашинное информационное обеспечение (макеты/экранные формы для ввода первичных данных в ПК или вывода результатной информации, структуры информационной базы: входных, выходных файлов, базы данных).

Среди всего многообразия средств в методологиях структурного анализа наиболее часто и эффективно применяемыми являются следующие:

- DFD (Data Flow Diagrams) – диаграммы потоков данных совместно со словарями данных и спецификациями процессов или миниспецификациями;
- ERD (Entity-Relationship diagrams) – диаграммы «сущность-связь».

### **2.1.1 Диаграмма потоков данных**

Для демонстрации как каждый процесс преобразует свои выходные данные в выходные, а также выявления отношения между этими процессами тестирование медицинских работников используется методология DFD (Data Flow Diagrams) – диаграммы потоков данных, которая представляет собой иерархию функциональных процессов, связанных потоками данных.

Основными компонентами диаграмм потоков данных являются:

1. внешняя сущность – представляет собой материальный объект или физическое лицо, являющиеся источником или приемником информации;
2. процесс – представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом;
3. накопитель данных это абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми;
4. поток данных определяет информацию, передаваемую через некоторое соединение от источника к приемнику [9].

На рисунке 11 построена контекстная диаграмма. Начальная контекстная диаграмма потоков данных содержит процесс тестирования медицинских работников (пользователей системы). Внешними сущностями являются: медицинские работники и Министерство здравоохранения Республики Хакасия.

Декомпозиция контекстной диаграммы представлена на рисунке 12.

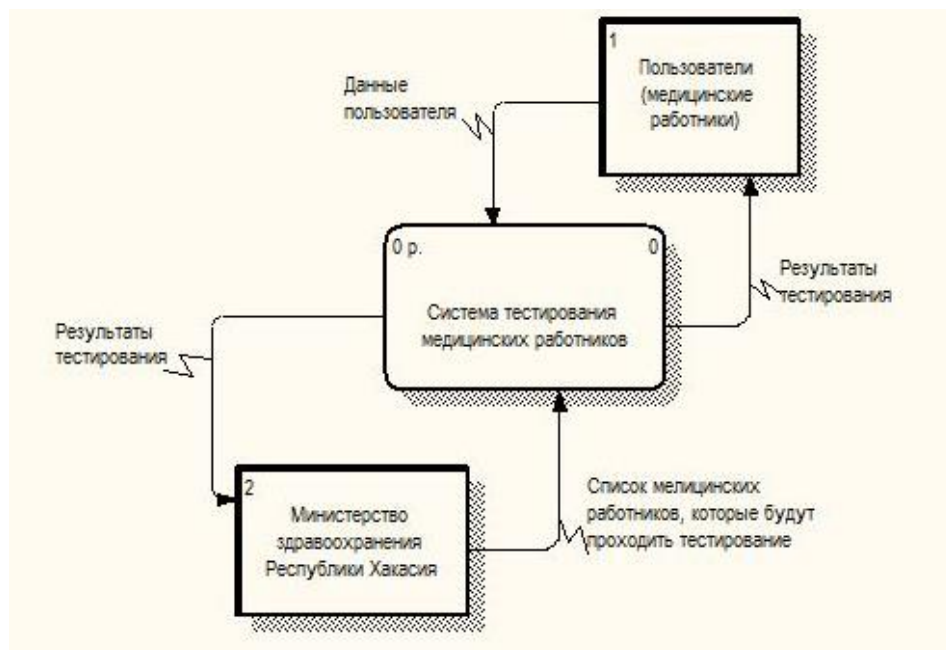


Рисунок 11 – Контекстная диаграмма в методологии DFD

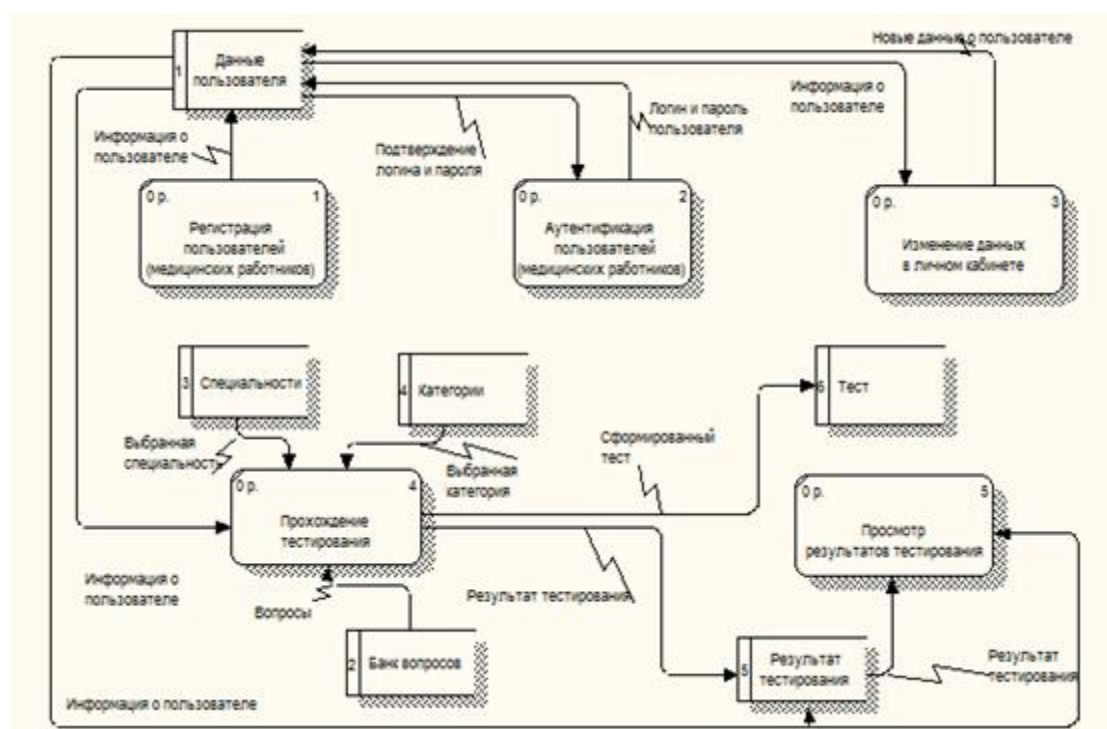


Рисунок 12 – Декомпозиция контекстной диаграммы в методологии DFD

В декомпозиции выделены следующие процессы: регистрация пользователей (медицинских работников), аутентификация пользователей

(медицинских работников), изменение данных пользователя, формирование теста, тестирование, просмотр результатов тестирования. В диаграмме представлены следующие накопители данных: данные пользователя, банк вопросов, специальности, категории, тест, результаты тестирования.

### 2.1.2 Диаграмма «сущность-связь»

Из ранее построенных диаграмм потоков данных, представленных на рисунках 11-12, выявлены накопители, которые будут составлять логическую схему базы данных системы тестирования.

При создании логической модели БД производится построение диаграммы ERD (Entity Relationship Diagram). ERD-диаграммы состоят из трех частей: сущностей, атрибутов и взаимосвязей.

Первым этапом является определение сущностей и атрибутов. В БД хранится информация о медицинских работниках, следовательно, сущностью является медицинский работник (таблица 5).

Таблица 5 – Атрибуты сущности «Медицинские работники»

Атрибут	Сущность
Номер	Уникальный номер для идентификации пользователя
Тип пользователя	Разграничение прав доступа
Логин	Логин для доступа в систему
Пароль	Пароль для доступа в систему в зашифрованном виде
Электронная почта	Адрес электронной почты
Фамилия	Фамилия пользователя
Имя	Имя пользователя
Отчество	Отчество пользователя
Группа специальностей	Группа специальностей, к которой относится специальность медицинского работника
Специальность	Специальность, по которой осуществляет свою деятельность медицинский работник
Категория	Квалификационная категория медицинского работника
Стаж работы	Продолжительность работы по трудовому договору и по определенной специальности

## Окончание таблицы 5

Место работы	Медицинское учреждение, где работает медицинский работник
Телефон	Номер телефона медицинского работника
Пробное тестирование	Отметка о прохождении пробного тестирования медицинским работником
Основное тестирование	Отметка о прохождении основного тестирования медицинским работником
hash	Уникальная для каждого пользователя строка, используемая для подтверждения регистрации
confirm	Отметка о прохождении медицинским работником процедуры подтверждения учетной записи
session	Уникальная для каждого пользователя переменная, используемая для идентификации сессии

В полученном списке существуют атрибуты, которые являются справочниками, поэтому выделяются в отдельные сущности. К таким атрибутам относятся: группа специальностей, специальности, категории, место работы. В таблицах 6, 7, 8, 9 представлены данные сущности.

Таблица 6– Атрибуты сущности «Группа специальностей»

Атрибут	Сущность
Номер группы специальностей	Уникальный номер для идентификации группы
Группа	Наименование группы специальностей

Таблица 7 – Атрибуты сущности «Специальности»

Атрибут	Сущность
Номер специальности	Уникальный номер для идентификации специальности
Номер группы специальности	Список специальностей зависит от группы специальностей
Специальность	Наименование специальности

Таблица 8– Атрибуты сущности «Категория»

Атрибут	Сущность
Номер категории	Уникальный номер для идентификации категории
Категория	Наименование категории



Таблица 9 – Атрибуты сущности «Место работы»

Атрибут	Сущность
Номер места работы	Уникальный номер для идентификации места работы
Медицинское учреждение	Наименование медицинского учреждения

Для формирования теста необходимо хранить набор вопросов и вариантов ответов к ним. Так как к одному вопросу относятся несколько вариантов ответов, хранятся они в разных сущностях (Таблица 10 – 11).

Таблица 10 – Атрибуты сущности «Вопросы»

Атрибут	Сущность
Номер	Уникальный номер для идентификации вопроса
Вопрос	Текст вопроса
Тип вопроса	Одиночный ответ или ответ с множественным выбором
Специальность	Специальность, к которой относится вопрос

Таблица 11 – Атрибуты сущности «Ответы»

Атрибут	Сущность
Номер	Уникальный номер для идентификации ответа
Ответ	Текст ответа
Номер вопроса	Уникальный номер вопроса, к которому относится ответ
Корректность ответа	Отметка о правильности варианта ответа

Так как существует зависимость вопроса от специальности медработников, но при этом один вопрос может использоваться в смежных специальностях, необходимо использовать промежуточную, связующую сущность «Вопросы по специальностям» (Таблица 12).

При прохождении тестирования в БД помимо результатов тестирования хранится информация о сформированном тесте. В таблице 13 представлена сущность «Тест» с атрибутами, 14 – сущность «Результаты теста».

Таблица 12 – Атрибуты сущности «Вопросы по специальностям»

Атрибут	Сущность
Номер	Идентификационный номер связи
Вопрос	Идентификационный номер вопроса
Специальность	Идентификационный номер специальности, к которой относится вопрос

Таблица 13 – Атрибуты сущности «Тест»

Атрибут	Сущность
Номер	Уникальный номер для идентификации теста
Номер вопроса	Уникальный номер вопроса
Ответ пользователя	Номер ответа, выбранный пользователем
Тип теста	Пробный или основной

Таблица 14 – Атрибуты сущности «Результаты теста»

Атрибут	Сущность
Номер записи	Идентификатор результата
Дата	Дата и время тестирования
Пользователь	Идентификатор пользователя, который походил тест
Тест	Идентификатор теста, сформированного ранее
Результат	Полученный результат тестирования (количество правильных ответов / количество вопросов в тесте)
Процент	Результат в процентах

На рисунке 12 представлена логическая схема данных, где все ключевые атрибуты находятся над горизонтальной чертой внутри рамки, изображающей сущность.

На диаграмме связь между сущностями имеет два вида: идентификационная связь один-ко многим (сплошная линия) и не идентифицирующая связь один-ко многим (пунктирная линия).

Идентифицирующая связь устанавливается между независимой (родительский) и зависимой (дочерний) сущностями. При установлении не идентифицирующей связи дочерняя сущность остается независимой, а атрибуты первичного ключа родительской сущности мигрируют в состав не ключевых компонентов родительской сущности.

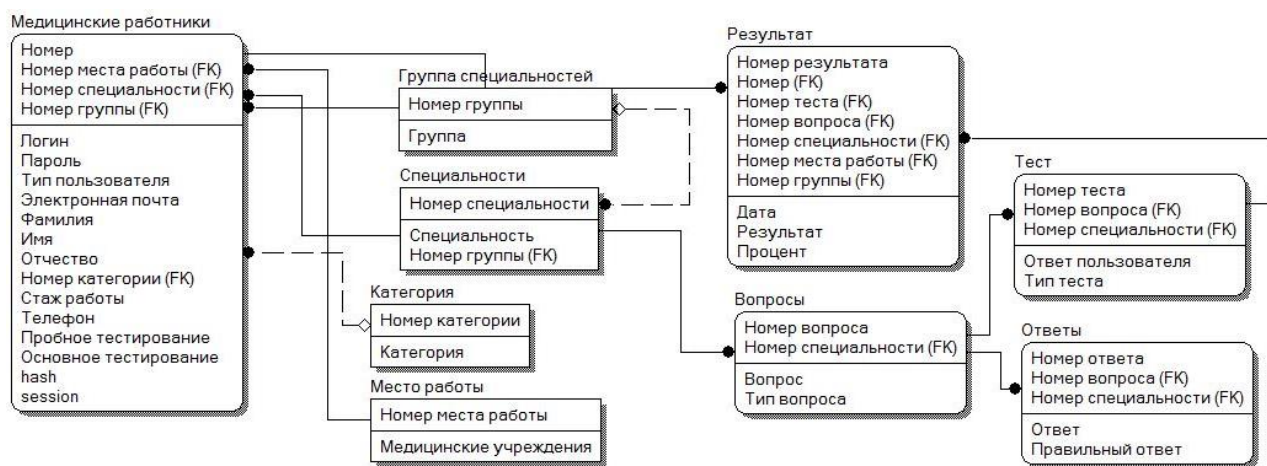


Рисунок 12 – Логическая модель данных, на основе методологии ER

Для переноса логической модели данных в СУБД создается физическая модель данных. При этом логическая модель трансформируется в физическую по следующему принципу: сущности становятся таблицами, атрибуты становятся столбцами, а ключи становятся индексами.

На рисунке 13 представлена модель, в которой скорректированы типы и размеры полей, введены правила валидации колонок, определяющие списки допустимых значений и значения по умолчанию.

Таким образом, проделав все вышеописанные действия, полученная модель БД проектируется в СУБД MySQL.

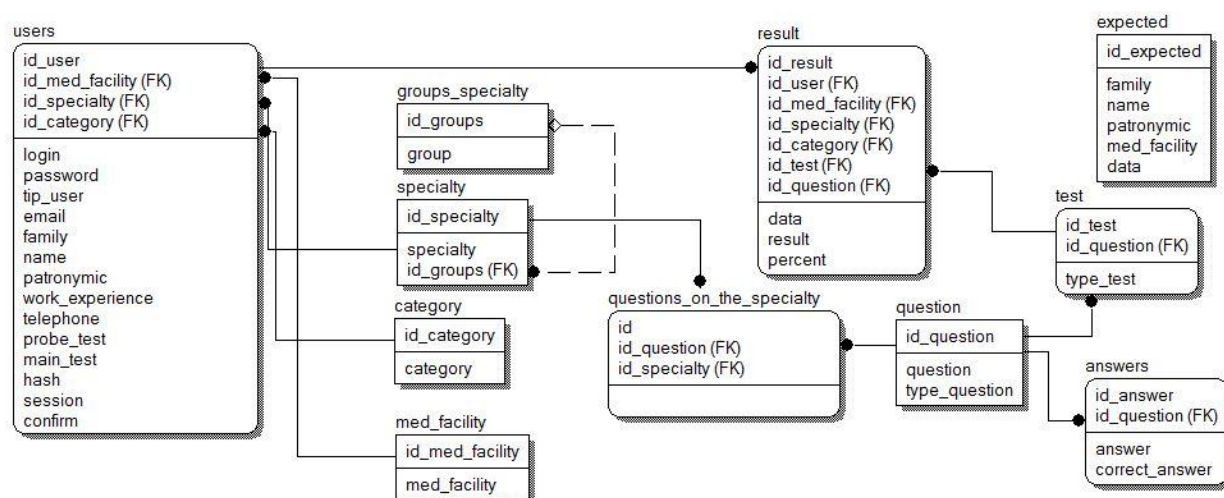


Рисунок 13 – Физическая модель данных, на основе методологии ER

## **2.2 Разработка модуля информационной системы, реализующего процесс аттестационного тестирования медицинских работников**

Для разработки модуля тестирования информационной системы необходимо создать базу данных и реализовать следующие функции:

- регистрация пользователя;
- авторизация пользователя;
- изменение личных данных пользователя;
- просмотр результатов тестирований пользователя;
- тестирование пользователя.

### **2.2.1 Файловая структура**

База данных представляет собой совокупность связанных таблиц, служащих для обеспечения работоспособности всей системы в целом.

База данных состоит из 11 таблиц, подлежащих добавлению, изменению и удалению информации:

- «users» - хранит всю личную информацию о пользователях системы;
- «questions», «answers» хранят банк вопросов и варианты ответов к ним;
- справочники «groups», «specialties», «categories», «medical\_facility» содержат информацию о группах медицинских специальностей, список самих специальностей, возможных категориях и список медицинских учреждений соответственно;
- «tests» - содержит список сформированных тестов;
- «test\_results» - информация о результатах тестирований;
- «questions\_specialties» - таблица, связующая вопросы и специальности;
- «expected» - список ожидаемых пользователей для тестирования.

Система тестирования поделена на три логических модуля:

- user – модуль пользователя (тестируемого), предназначен для регистрации, авторизации пользователей, формирования тестов из банка

вопросов, прохождения тестирования, подсчет результатов тестирования;

- admin – модуль администратора системы, предназначен для администрирования всей системы в целом, добавления, редактирования, удаления вопросов, подтверждения, блокировки, разблокировки пользователей, обнуления попыток тестирований, редактирования справочников;

- moderator – модуль представителя Министерства здравоохранения РХ, обладает ограниченными функциями admin: наполнение банка вопросов по специальностям, просмотр результатов тестирований, заполнения списка ожидаемых пользователей для тестирования.

Каждый модуль представляет собой набор скриптов, каскадных таблиц стилей (CSS) и веб-страниц с блоками кода PHP, расположенных в корневом каталоге системы тестирования.

Точкой входа системы является файл «index.php», представляющий собой раздел сайта «Главная».

Каждая страница сайта выделена в отдельный PHP файл. Полный код всех страниц системы представлен в приложении Б.

Все функции системы расположены в файле .../functions.php, при нажатии submit кнопок в зависимости от данных формы вызываются соответствующие функции. Описаны функции извлечения, изменения, добавления и удаления данных в БД, обработки переменных и объявления сессионных переменных.

Все страницы сайта логически можно разделить на три части:

- header – верхняя часть (шапка) сайта, в которой расположен логотип, заголовок сайта, навигационное меню в виде ссылок, форма авторизации (рисунок 14);

- content – основная часть сайта, будет меняться в зависимости от назначения страницы, на рисунке 15 – выбор категории и специальности перед тестированием;

- footer – нижняя часть (подвал) сайта, где находится информация о собственнике системы и номер телефона технической поддержки (рисунок 16).

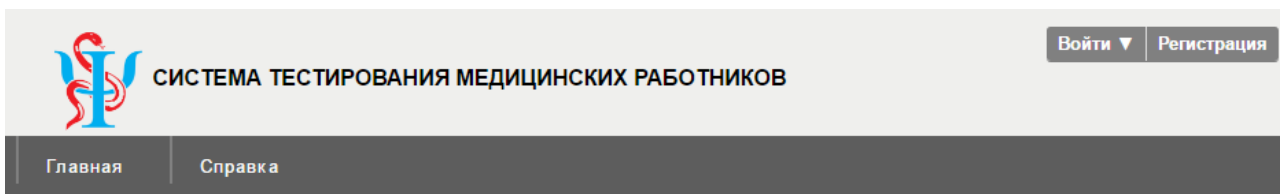


Рисунок 14 – Верхняя часть сайта

Для неавторизированных пользователей доступны следующие страницы:

- index.php – главная страница сайта;
- reference.php – раздел «Справка» сайта;
- tech.php – раздел техподдержки сайта;
- returnpass.php – страница восстановления пароля;
- register.php – страница регистрации пользователей.

Для авторизованных пользователей дополнительно доступны:

- kabinet.php – страница личного кабинета пользователей;
- testp.php – форма выбора специальности для тестирования, типа теста;
- test.php – страница самого теста.

---

**Система предназначена для прохождения аттестационного тестирования медицинскими работниками.**

Для тестирования необходимо зарегистрироваться или авторизоваться (если уже есть учетная запись) в системе.

В СПРАВКЕ описано, как работать с системой пользователю.

Если Вы не можете войти в систему, обратитесь в техническую поддержку.

**Посмотреть график заседаний  
аттестационных комиссий,  
документацию к аттестации ВРАЧЕЙ**

**Посмотреть график заседаний  
аттестационных комиссий,  
документацию к аттестации  
СРЕДНЕГО МЕДИЦИНСКОГО  
ПЕРСОНАЛА**

---

Рисунок 15 – Основная часть сайта

Рисунок 16 – Нижняя часть сайта

Навигационное меню в шапке страницы не статично, в зависимости от авторизации пользователя выводится различный набор ссылок. Для неавторизованного посетителя, вид меню представлен на рисунке 14, для авторизованного – на рисунке 17.

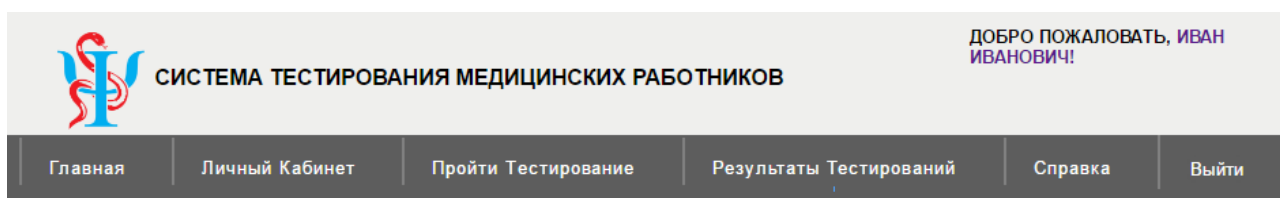


Рисунок 17 – Верхняя часть сайта авторизованного пользователя

### 2.2.2 Создание функции регистрации пользователя

Работа пользователя с системой начинается с регистрации. Ссылка на страницу регистрации расположена в шапке сайта (рис. 14).

Помимо регистрации данная функция должна предусматривать подтверждение учетной записи (через электронную почту) и подтверждение медицинского работника (через список ожидаемых пользователей для тестирования).

При регистрации необходимо заполнить форму регистрации, указав личные данные пользователя и дать свое согласие на обработку персональных данных (ПДн), отметив соответствующее поле (рис. 18-19).

Во избежание ввода некорректных данных пользователем при указании специальности, места работы, врачебной категории, вместо текстовых полей используются выпадающие списки, данные в которые подгружаются из

соответствующих таблиц БД.

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

Войти ▼ Регистрация

Главная Справка

Регистрация пользователя

Логин

Пароль

Подтверждение пароля

Электронная почта

Фамилия

Имя

Отчество

Группа специальностей

Рисунок 18 – Фрагмент страницы регистрации пользователей

Группа специальностей

Специальность

Стаж работы по специальности

Место работы

Категория

Телефон +7

☐ Я даю свое согласие на автоматизированную обработку предоставляемых персональных данных на основании Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных»

Регистрация

© 2017 ГКУЗ РХ "РМИАЦ" Техническая поддержка: 8 (3902) 295-025

Рисунок 19 – Фрагмент страницы регистрации пользователей



Также, для удобства выбора, специальности разделены на группы. При выборе определенной группы меняется список специальностей.

После заполнения формы регистрации, при нажатии кнопки «Регистрация» выполняются проверки на заполнение всех полей и совпадение паролей, проверка на отсутствие зарегистрированных пользователей с таким же логином и адресом электронной почты. При успешном прохождении всех проверок на указанный адрес электронной почты высылается письмо со ссылкой для подтверждения учетной записи. В противном случае выводится сообщение об ошибке.

Для завершения регистрации необходимо подтвердить учетную запись, перейдя по ссылке из письма (рис. 20). Подтверждение пользователя состоит из двух этапов: подтверждение регистрации и подтверждение пользователя как медицинского работника (проверка наличия пользователя с такими ФИО, МО, специальностью в списках пользователей, ожидающих тестирования).

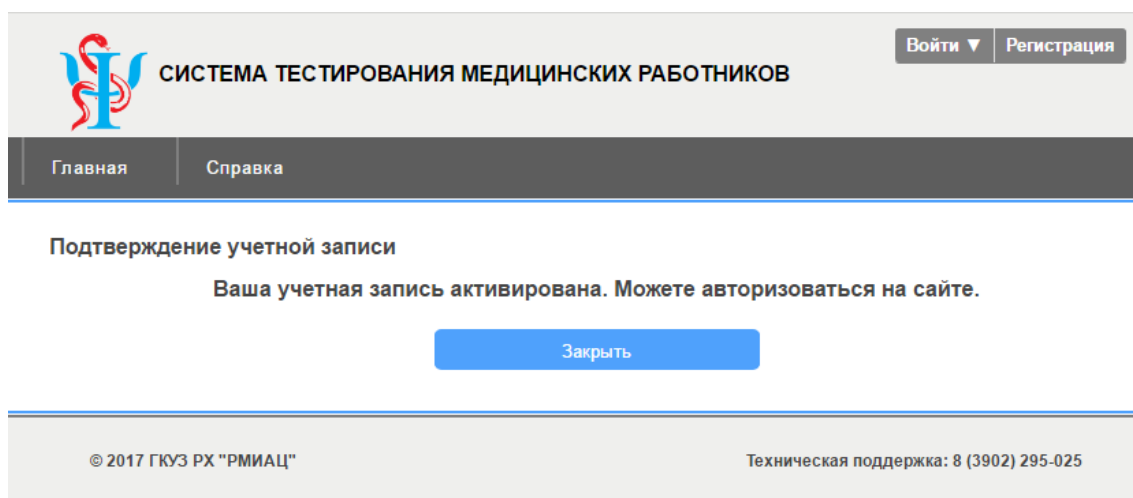


Рисунок 20 – Страница подтверждения регистрации пользователей

### 2.2.3 Создание функции авторизация пользователя

После завершения регистрации и успешной активации учетной записи, пользователь может пройти авторизацию в системе.

Помимо авторизации пользователя данная функция должна осуществлять восстановление забытого пароля.

Для авторизации необходимо ввести логин и пароль в форму входа на сайт (рис. 21). При нажатии кнопки «Войти» осуществляется вызов функции входа, которая для указанного пользователем логина выполняет проверку совпадения пароля, введенного пользователем, и пароля в БД.

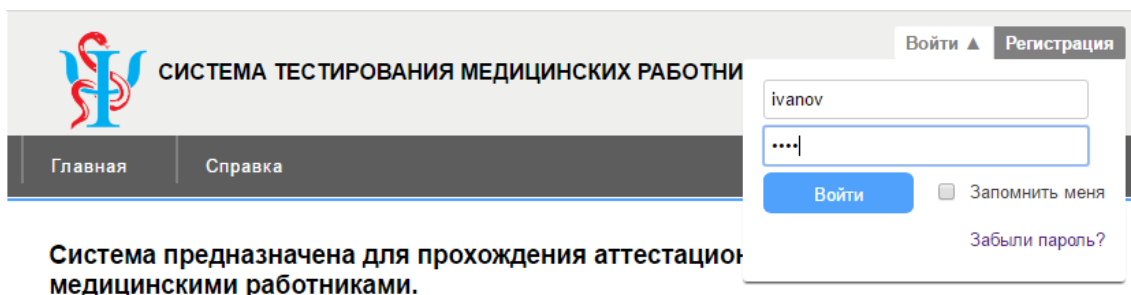


Рисунок 21 – Окно авторизации пользователей

Ссылка на страницу восстановления пароля также расположена на форме авторизации (рис. 21). Для восстановления пароля необходимо указать адрес электронной почты, к которому привязана учетная запись. Так как на один почтовый ящик может быть зарегистрирован только один пользователь, никаких дополнительных данных вводить не требуется. Вид страницы восстановления паролей представлен на рисунке 22.

#### 2.2.4 Создание личного кабинета пользователя

Личный кабинет предусматривает отображение личных данных авторизованного пользователя, предоставляет возможность их изменения при необходимости, а также выводит для просмотра все результаты тестирований.

При успешной авторизации, пользователь автоматически перенаправляется на страницу личного кабинета, где для просмотра выведены все его данные из БД (рис. 23).

The screenshot shows the password recovery interface. At the top, there is a logo on the left and navigation links 'Войти' and 'Регистрация' on the right. Below the header is a dark navigation bar with 'Главная' and 'Справка'. The main content area is titled 'Восстановление пароля' and contains the instruction 'Введите адрес почтового ящика для восстановления пароля'. There is a text input field labeled 'e-mail' and a blue button labeled 'Восстановить'. The footer contains copyright information '© 2017 ГКУЗ РХ "РМИАЦ"' and technical support contact '8 (3902) 295-025'.

Рисунок 22 – Страница восстановления паролей

The screenshot shows the personal cabinet page for a user named 'Иванов Иван Иванович'. The header includes the logo, system name, and a greeting 'ДОБРО ПОЖАЛОВАТЬ, ИВАН ИВАНОВИЧ!'. The navigation bar has links for 'Главная', 'Личный Кабинет', 'Пройти Тестирование', 'Результаты Тестирований', 'Справка', and 'Выйти'. The main content area is titled 'Личный кабинет' and displays the user's name. Below this, it lists personal data: 'Место работы: ГБУЗ РХ "Сорская городская больница"', 'Специальность: Сестринское дело', 'Стаж: 12', and 'Врачебная категория: Первая'. Contact information is also shown: 'Электронная почта: ivanov@mail.ru' and 'Телефон: 9332006577'. A blue button labeled 'Изменить' is at the bottom of the data section. The footer is identical to the previous screenshot.

Рисунок 23 – Страница личного кабинета

При нажатии кнопки изменения данных открывается страница редактирования данных пользователя (рис. 24).

Для просмотра результатов тестирований необходимо выбрать соответствующий пункт главного меню. Вид страницы результатов

тестирований представлен на рисунке 25.

Для предоставления справочной информации посетителям сайта, не зависимо от авторизации, разработан раздел «Справка», в котором описаны общие принципы работы с системой, указана информация о технической поддержке системы.

Изменить личные данные

Электронная почта	<input type="text" value="ivanov@mail.ru"/>
Пароль	<input type="password" value="Пароль"/>
Подтверждение пароля	<input type="password" value="Повторите пароль"/>
Фамилия	<input type="text" value="Иванов"/>
Имя	<input type="text" value="Иван"/>
Отчество	<input type="text" value="Иванович"/>
Специальность	<input type="text" value="Сестринское дело"/>
Стаж работы по специальности	<input type="text" value="12"/>
Место работы	<input type="text" value="ГБУЗ РХ " бол"="" городская="" сорская=""/>
Категория	<input type="text" value="Первая"/>
Телефон +7	<input type="text" value="9332006577"/>
<input type="button" value="Сохранить"/>	

© 2017 ГКУЗ РХ "РМИАЦ"

Техническая поддержка: 8 (3902) 295-025

Рисунок 24 – Фрагмент страницы редактирования личных данных

### 2.2.5 Создание функции тестирования

Функция тестирования подразумевает формирование теста (случайный выбор 70 тестовых вопросов с вариантами ответов из банка вопросов), временное ограничение на прохождение теста – 95 минут, проверка верности ответов, данных пользователем, подсчет результатов теста, запись их в БД и вывод на просмотр пользователю.

Перед началом тестирования пользователю необходимо выбрать специальность и категорию, по которым будет производиться тестирование (рис. 26). После выбора открывается страница теста, сформированного из банка вопросов произвольным образом.

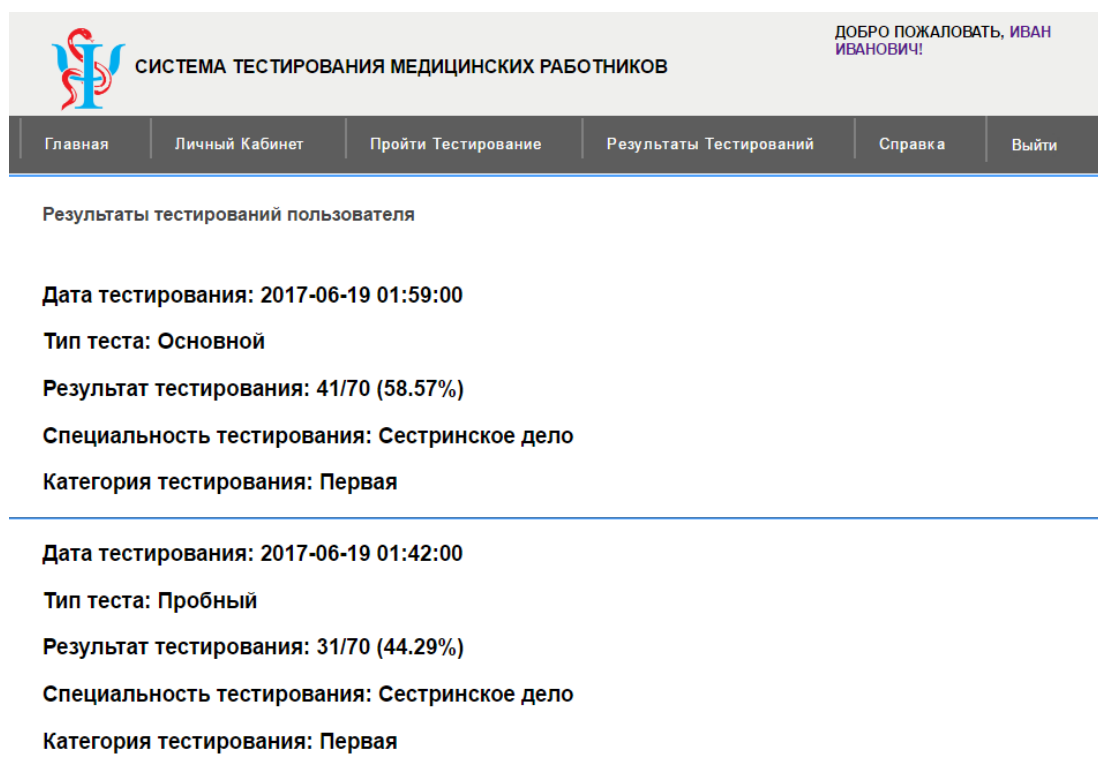


Рисунок 25 – Фрагмент страницы просмотра результатов тестирований

Для всех вопросов теста строится пагинация (вывод ограниченной части информации). Из всего массива вопросов постоянно отображается только один, остальные скрыты. При нажатии на ссылку вопроса отображаемый вопрос скрывается, а выводится выбранный. Дополнительно подсвечиваются уже просмотренные пользователем вопросы. Данные возможности реализована через методы `removeClass`, `addClass` библиотеки `jQuery`, языка `JavaScript` и стили элементов.

Реализована навигация по вопросам – вывод номеров вопросов в виде ссылок. Постоянно отображаются все вопросы, для более удобной навигации и информативности.

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, ИВАН ИВАНОВИЧ!

Главная | Личный Кабинет | Пройти Тестирование | Результаты Тестирований | Справка | Выйти

Пройти Тестирование

Укажите специальность и категорию для тестирования

Группа специальностей Выберите группу ▼

Специальность Выберите специальность ▼

☐ Пробный тест

Начать тестирование

© 2017 ГКУЗ РХ "РМИАЦ" Техническая поддержка: 8 (3902) 295-025

Рисунок 26 – Выбор параметров тестирования

На рисунке 27 представлен пример теста из трех вопросов, с ограничением времени в 30 секунд.

Завершение теста возможно двумя способами, при нажатии соответствующей кнопки пользователем или по истечении времени отведенного на тестирование.

В обоих случаях осуществляется проверка данных пользователем ответов, подсчет количества не отвеченных вопросов, не верных и верных ответов, процент верных ответов, в зависимости от которого подводится итог тестирования (тестирование пройдено / тестирование не пройдено). Окно результатов тестирования можно увидеть на рисунке 28.

### 2.3 Обеспечение информационной безопасности

Обеспечение информационной безопасности – одна из важнейших частей разработки информационной системы.

В ходе разработки системы реализована защита от SQL-инъекций. Перед отправкой введенных пользователем данных они преобразуются специальными

функциями (mysqli\_real\_escape\_string, trim). Производится проверка содержания всех полей форм на стороне пользователя средствами HTML5.

**Вопросы выбранной специальности**

Всего вопросов: 70

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

Наличие на инструментах остатков моющих средств проверяется с помощью пробы

☐

Азопирамовой

☐

Амидопириновой

☐

Бензидиновой☐


Закончить тест

94:51

© 2017 ГКУЗ РХ "РМИАЦ"

Техническая поддержка: 8 (3902) 295-025

Рисунок 27 – Страница теста



СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

Главная

Личный Кабинет

Пройти Тестирование

Результаты Тестирований

Справка

Выйти

Всего вопросов: 70

Отвечено верно: 41

Отвечено не верно: 29

Осталось не отвеченными: 0

Верных ответов: 58.57 %

ТЕСТИРОВАНИЕ НЕ ПРОЙДЕНО!

Закрыть

© 2017 ГКУЗ РХ "РМИАЦ"

Техническая поддержка: 8 (3902) 295-025

Рисунок 28 – Окно результатов тестирования

Для хранения паролей, кода подтверждения учетной записи и сессионной переменной перед их отправкой на сервер указанные данные хешируются однонаправленным шифром md5.

Также используется алгоритм разграничения прав доступа. Реализована функция проверки авторизации пользователя, которая добавлена в начало каждой страницы, доступ к которой возможен только авторизованному пользователю.

Во избежание доступа к тестовым вопросам посторонних лиц, организована проверка регистрируемых пользователей в списке ожидаемых пользователей для тестирования. Таким образом, доступ в систему имеют только те пользователи, которые подали заявление на прохождение аттестации в Министерство здравоохранения, и добавлены в соответствующий список.

Основная учетная запись СУБД MySQL защищена паролем.

Помимо этого, контрольно-пропускной режим позволит избежать физического несанкционированного доступа к информационной системе предприятия.

При внедрении системы, для защиты от потери данных необходимо настроить резервное копирование базы данных MySQL.

## **2.4 Контрольный пример**

Для проверки работоспособности разработанной информационной системы аттестационного тестирования медицинских работников проведены регистрация, подтверждение, тестирование пользователя со следующими данными.

ФИО: Семенова Мария Александровна.

Место работы: ГБУЗ РХ «Бейская районная больница».

Специальность: Сестринское дело.

Стаж работы по специальности: 11 лет.

Врачебная категория: Вторая.



Контактный телефон: 89230000000.

Адрес электронной почты: sma@mail.ru.

Логин: semenova.

Пароль: 1234.

Окно регистрации данного пользователя представлено на рисунке 29, 31, 33, 35, .

Вывод сообщений об ошибках предусмотрен в следующих случаях: пользователь не указал место работы, врачебную категорию, специальность (рис. 32, 34, 30); при несовпадении введенных паролей (рис. 38); при попытке зарегистрировать пользователя с уже существующим логином (рис. 36), адресом электронной почты (рис. 37). После вывода сообщения пароли и некорректное поле очищаются, остальные поля сохраняют введенные пользователем данные.

Вывод сообщения об успешной регистрации представлен на рисунке 39.

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

Войти ▼ Регистрация

Главная Справка

Регистрация пользователя

Логин

Пароль

Подтверждение пароля

Электронная почта

Фамилия

Имя

Отчество

Группа специальностей

Специальность

- Медицинский и фармацевтический персонал
- Средний медицинский персонал

Рисунок 29 – Выбор группы специальностей при регистрации пользователя

## Выберите Специальность!

### Регистрация пользователя

Логин	<input type="text" value="semenova"/>
Пароль	<input type="text" value="Пароль"/>
Подтверждение пароля	<input type="text" value="Повторите пароль"/>
Электронная почта	<input type="text" value="sma@mail.ru"/>
Фамилия	<input type="text" value="Семенова"/>
Имя	<input type="text" value="Мария"/>
Отчество	<input type="text" value="Александровна"/>

Рисунок 30 – Ошибка при регистрации пользователя

### Регистрация пользователя

Логин	<input type="text" value="semenova"/>
Пароль	<input type="password" value="...."/>
Подтверждение пароля	<input type="password" value="...."/>
Электронная почта	<input type="text" value="sma@mail.ru"/>
Фамилия	<input type="text" value="Семенова"/>
Имя	<input type="text" value="Мария"/>
Отчество	<input type="text" value="Александровна"/>
Группа специальностей	<div> <div>Акушерское дело</div> <div>Гигиеническое воспитание</div> <div>Сестринское дело</div> <div>Зубные техники</div> <div>Медицинский массаж</div> <div>Эпидемиология</div> <div>Операционное дело</div> </div>
Специальность	<input type="text" value="Акушерское дело"/>

Рисунок 31 – Выбор специальности при регистрации пользователя

## Выберите Место работы!

### Регистрация пользователя

Логин	<input type="text" value="semenova"/>
Пароль	<input type="password" value="Пароль"/>
Подтверждение пароля	<input type="password" value="Повторите пароль"/>
Электронная почта	<input type="text" value="sma@mail.ru"/>
Фамилия	<input type="text" value="Семенова"/>
Имя	<input type="text" value="Мария"/>
Отчество	<input type="text" value="Александровна"/>

Рисунок 32 – Ошибка при регистрации пользователя


Группа специальностей	<div style="border: 1px solid #ccc; padding: 2px;">             ГБУЗ РХ «Аскизская межрайонная больница»  <b>ГБУЗ РХ «Бейская районная больница»</b>              ГБУЗ РХ «Белоярская районная больница»              ГБУЗ РХ «Копьевская районная больница»              ГБУЗ РХ «Республиканская клиническая больница им. Г.Я. Ремишевской»              ГБУЗ РХ «Республиканская клиническая инфекционная больница»              ГБУЗ РХ «Республиканская клиническая офтальмологическая больница имени Н. М. Одежкина»              ГБУЗ РХ «Таштыпская районная больница»              ГБУЗ РХ "Сорская городская больница"           </div>
Специальность	
Стаж работы по специальности	
Место работы	<div style="border: 1px solid #ccc; padding: 2px;">Выберите место работы ▼</div>
Категория	<div style="border: 1px solid #ccc; padding: 2px;">Выберите категорию ▼</div>
Телефон +7	<input type="text" value="Телефон"/>

☐ Я даю свое согласие на автоматизированную обработку предоставляемых персональных данных на основании Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных»

Регистрация

© 2017 ГКУЗ РХ "РМИАЦ"
Техническая поддержка: 8 (3902) 295-025

Рисунок 33 – Выбор места работы при регистрации пользователя



СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

Войти ▾Регистрация

ГлавнаяСправка

Выберите Категорию!

Регистрация пользователя

Логин

semenova

Пароль

Пароль

Подтверждение пароля

Повторите пароль

Электронная почта

sma@mail.ru

Фамилия

Семенова

Имя

Мария

Отчество

Александровна

Группа специальностей

Средний медицинский персонал ▾

Рисунок 34 – Ошибка при регистрации пользователя

Место работы

ГБУЗ РХ «Бейская районная бол» ▾

Категория

Выберите категорию ▾  
Выберите категорию  
Первая  
Вторая  
Высшая

Телефон +7

☐ Я даю свое согласие на автоматизированную обработку предоставляемых персональных данных на основании Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных»

Регистрация

© 2017 ГБУЗ РХ "РМИАЦ"Техническая поддержка: 8 (3902) 295-025

Рисунок 35 – Выбор врачебной категории при регистрации пользователя

Пользователь с таким логином уже существует!

Регистрация пользователя

Логин	<input type="text" value="Логин"/>
Пароль	<input type="text" value="Пароль"/>
Подтверждение пароля	<input type="text" value="Повторите пароль"/>
Электронная почта	<input type="text" value="sma@mail.ru"/>
Фамилия	<input type="text" value="Семенова"/>
Имя	<input type="text" value="Мария"/>
Отчество	<input type="text" value="Александровна"/>


Рисунок 36 – Ошибка при регистрации пользователя

Пользователь с таким e-mail уже существует!

Регистрация пользователя

Логин	<input type="text" value="semenova"/>
Пароль	<input type="text" value="Пароль"/>
Подтверждение пароля	<input type="text" value="Повторите пароль"/>
Электронная почта	<input type="text" value="Электронная почта"/>
Фамилия	<input type="text" value="Семенова"/>
Имя	<input type="text" value="Мария"/>
Отчество	<input type="text" value="Александровна"/>

Рисунок 37 – Ошибка при регистрации пользователя

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

Войти ▼Регистрация

ГлавнаяСправка

Вы неправильно подтвердили пароль!

Регистрация пользователя

Логинsemenova

ПарольПароль

Подтверждение пароляПовторите пароль


Электронная почтаsma@mail.ru

ФамилияСеменова

ИмяМария

ОтчествоАлександровна

Рисунок 38 – Ошибка при регистрации пользователя

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

Войти ▼Регистрация

ГлавнаяСправка

Вы успешно зарегистрировались на сайте. Для подтверждения регистрации Вам на почту отправлено письмо с инструкциями.

Регистрация пользователя

ЛогинЛогин

ПарольПароль

Рисунок 39 – Сообщение об успешной регистрации пользователя

После регистрации система автоматически формирует переменную hash, которая будет использоваться для подтверждения регистрации. При подтверждении, сообщение об ошибке выводится в случаях отсутствия пользователя в списке ожидаемых тестируемых (рис. 42) и отсутствия зарегистрированного пользователя с указанным полем hash (рис. 41).

Авторизация представлена на рисунке 43.

В личном кабинете пользователя (рис. 44) указаны все данные, введенные при регистрации, которые при необходимости можно изменить (рис. 45).

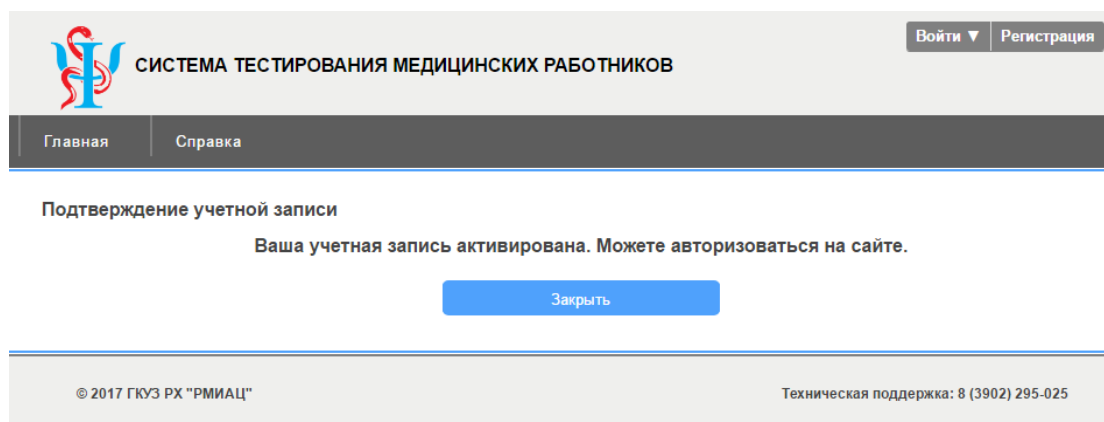


Рисунок 40 – Успешное подтверждение регистрации пользователя

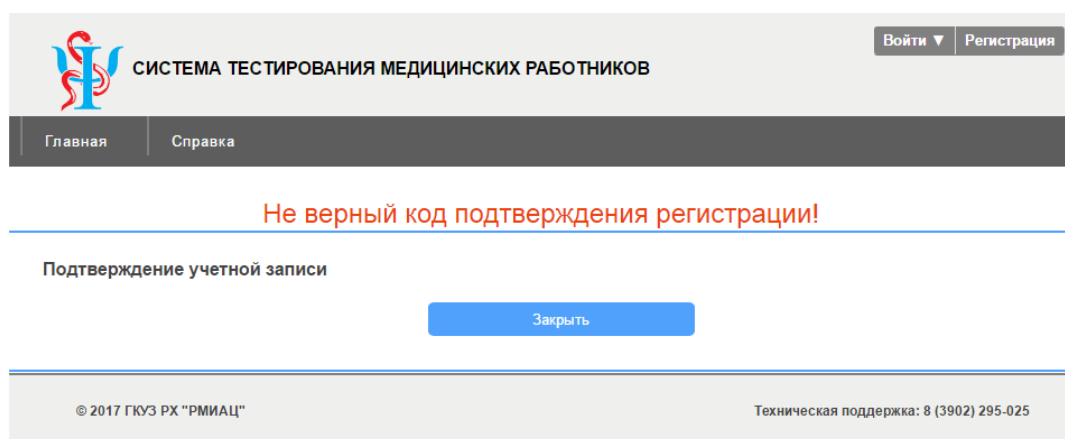


Рисунок 41 – Ошибка при подтверждении регистрации пользователя

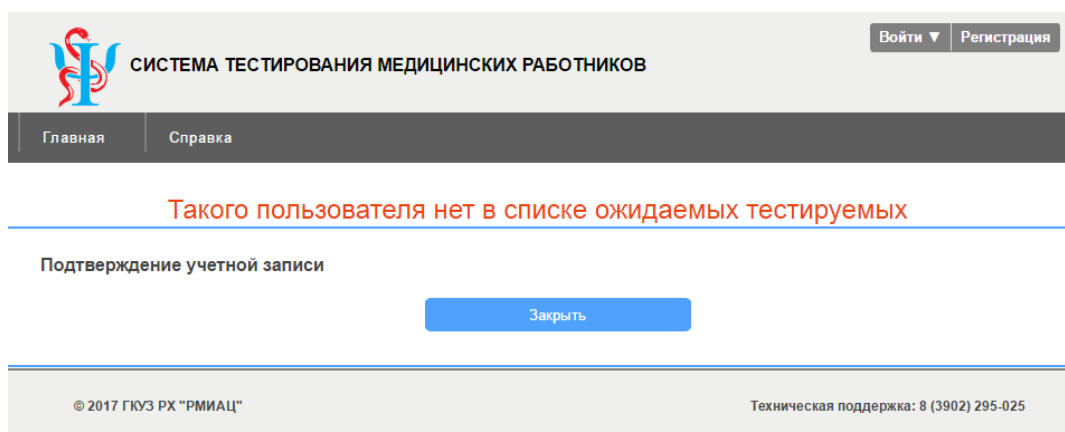


Рисунок 42 – Ошибка при подтверждении регистрации пользователя

**СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ**

Главная Справка

Войти ▲ Регистрация

semenova

....

Войти ☐ Запомнить меня

[Забыли пароль?](#)

**Система предназначена для прохождения аттестационного тестирования работниками.**

Для тестирования необходимо зарегистрироваться или авторизоваться (если уже есть учетная запись) в системе.

В СПРАВКЕ описано, как работать с системой пользователю.

Если Вы не можете войти в систему, обратитесь в техническую поддержку.

**Посмотреть график заседаний аттестационных комиссий, документацию к аттестации ВРАЧЕЙ**

**Посмотреть график заседаний аттестационных комиссий, документацию к аттестации СРЕДНЕГО МЕДИЦИНСКОГО ПЕРСОНАЛА**

© 2017 ГКУЗ РХ "РМИАЦ" Техническая поддержка: 8 (3902) 295-025

Рисунок 43 – Форма авторизации в системе

**СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ**

ДОБРО ПОЖАЛОВАТЬ, **МАРИЯ АЛЕКСАНДРОВНА!**

Главная Личный Кабинет Пройти Тестирование Результаты Тестирований Справка Выйти

**Личный кабинет**

**Семенова Мария Александровна**

**Место работы: ГБУЗ РХ «Бейская районная больница»**

**Специальность: Сестринское дело**

**Стаж: 11**

**Врачебная категория: Вторая**

**Электронная почта: sma@mail.ru**

**Телефон: 9230000000**

**Изменить**

Рисунок 44 – Страница личного кабинета пользователя



---

Изменить личные данные

Электронная почта

Пароль

Подтверждение пароля

Фамилия

Имя

Отчество

Специальность

Стаж работы по специальности

Место работы

Категория

Телефон +7

---

© 2017 ГКУЗ РХ "РМИАЦ"


Техническая поддержка: 8 (3902) 295-025

Рисунок 45 – Страница изменения данных пользователя

Перед тестированием необходимо выбрать специальность, в зависимости от которой формируется тест и указать тип теста. Пользователь не может: приступить к основному тесту, не пройдя пробный (рис. 48); начать тестирование не выбрав специальность(рис. 46); пройти пробный тест дважды (рис. 51). На основной тест дается одна попытка (рис. 56), дополнительная – после подачи соответствующего заявления предоставляется администратором системы.

Страницы пробного теста и результата тестирования представлены на рисунках 49 и 50 соответственно. Основного теста – рисунках 52 – 54. При тестировании идет учет отведенного времени, при окончании которого тестирование автоматически завершается (рис. 53), проверяется правильность данных пользователем ответов (рис. 54).

В личном кабинете хранятся результаты всех тестирований пользователя (рис. 55).



СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, МАРИЯ  
АЛЕКСАНДРОВНА!

ГлавнаяЛичный КабинетПройти ТестированиеРезультаты ТестированийСправкаВыйти

Не выбрана специальность

Пройти Тестирование

Укажите специальность и категорию для тестирования

Группа специальностейВыберите группу


СпециальностьВыберите специальность

☐ Пробный тест

Начать тестирование

© 2017 ГКУЗ РХ "РМИАЦ"Техническая поддержка: 8 (3902) 295-025

Рисунок 46 – Ошибка при выборе параметров тестирования



СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, МАРИЯ  
АЛЕКСАНДРОВНА!

ГлавнаяЛичный КабинетПройти ТестированиеРезультаты ТестированийСправкаВыйти

Пройти Тестирование

Укажите специальность и категорию для тестирования


Группа специальностейСредний медицинский персонал

Специальность

Акушерское дело  
Акушерское дело  
Гигиеническое воспитание  
Сестринское дело  
Зубные техники  
Медицинский массаж  
Эпидемиология  
Операционное дело

© 2017 ГКУЗ РХ "РМИАЦ"Техническая поддержка: 8 (3902) 295-025

Рисунок 47 – Выбор параметров тестирования

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, МАРИЯ  
АЛЕКСАНДРОВНА!

[Главная](#) | [Личный Кабинет](#) | [Пройти Тестирование](#) | [Результаты Тестирований](#) | [Справка](#) | [Выйти](#)

Вы не прошли пробный тест

Пройти Тестирование

Укажите специальность и категорию для тестирования

Группа специальностей


Специальность

☐ Пробный тест

Начать тестирование

© 2017 ГКУЗ РХ "РМИАЦ" | Техническая поддержка: 8 (3902) 295-025

Рисунок 48 – Ошибка при выборе параметров тестирования

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, МАРИЯ  
АЛЕКСАНДРОВНА!

[Главная](#) | [Личный Кабинет](#) | [Пройти Тестирование](#) | [Результаты Тестирований](#) | [Справка](#) | [Выйти](#)

Вопросы выбранной специальности

Всего вопросов: 15

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

Частота проведения генеральной уборки в процедурных и перевязочных кабинетах

☐ 1 раз в 3 дня

☐ 1 раз в 7 дней


☐ 1 раз в 10 дней

☐ 1 раз в месяц

Закончить тест

94:49

Рисунок 49 – Первый вопрос пробного тестирования



СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, МАРИЯ АЛЕКСАНДРОВНА!

[Главная](#)
[Личный Кабинет](#)
[Пройти Тестирование](#)
[Результаты Тестирований](#)
[Справка](#)
[Выйти](#)

Всего вопросов: 15

Отвечено верно: 5

Отвечено не верно: 10

Осталось не отвеченными: 0

Верных ответов: 33.33 %


**ТЕСТИРОВАНИЕ НЕ ПРОЙДЕНО!**

Заккрыть

© 2017 ГКУЗ РХ "РМИАЦ"

Техническая поддержка: 8 (3902) 295-025

Рисунок 50 – Результат пробного тестирования



СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, МАРИЯ АЛЕКСАНДРОВНА!

[Главная](#)
[Личный Кабинет](#)
[Пройти Тестирование](#)
[Результаты Тестирований](#)
[Справка](#)
[Выйти](#)

Вы уже прошли пробный тест

Пройти Тестирование

Укажите специальность и категорию для тестирования

Группа специальностей

Специальность

☐ Пробный тест

Начать тестирование

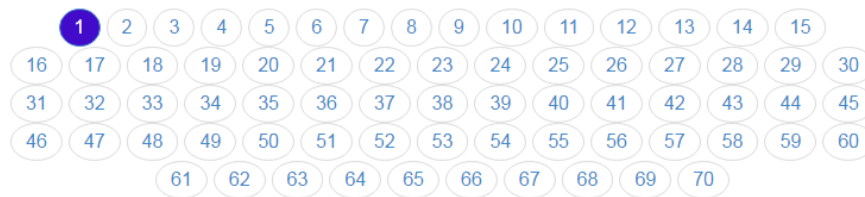
© 2017 ГКУЗ РХ "РМИАЦ"

Техническая поддержка: 8 (3902) 295-025

Рисунок 51 – Ошибка при выборе параметров тестирования

## Вопросы выбранной специальности

Всего вопросов: 70



Личная дистанция при повседневном общении медицинской сестры с пациентами, коллегами

- ☐ 0-45см
- ☐ 45-120см
- ☐ 120-400см
- ☐ 400-550см

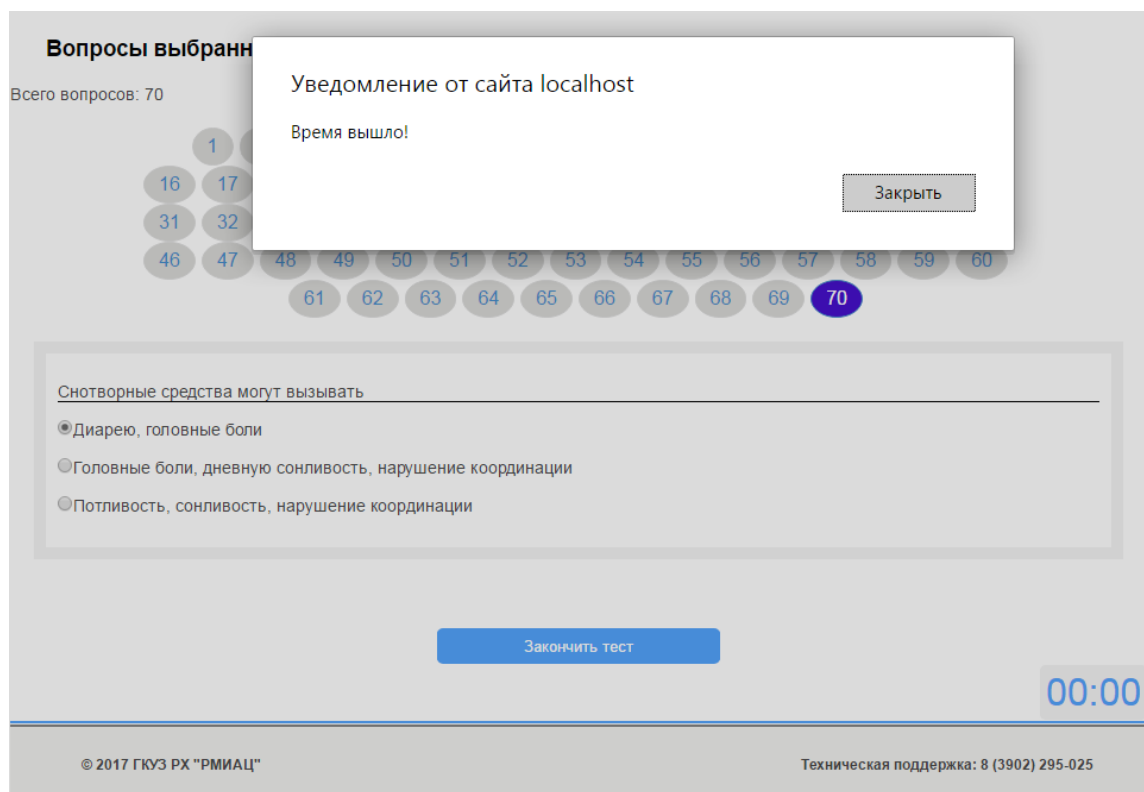
Закончить тест

94:46

© 2017 ГКУЗ РХ "РМИАЦ"

Техническая поддержка: 8 (3902) 295-025

## Рисунок 52 – Первый вопрос основного теста



## Рисунок 53 – Заключительный вопрос сформированного теста

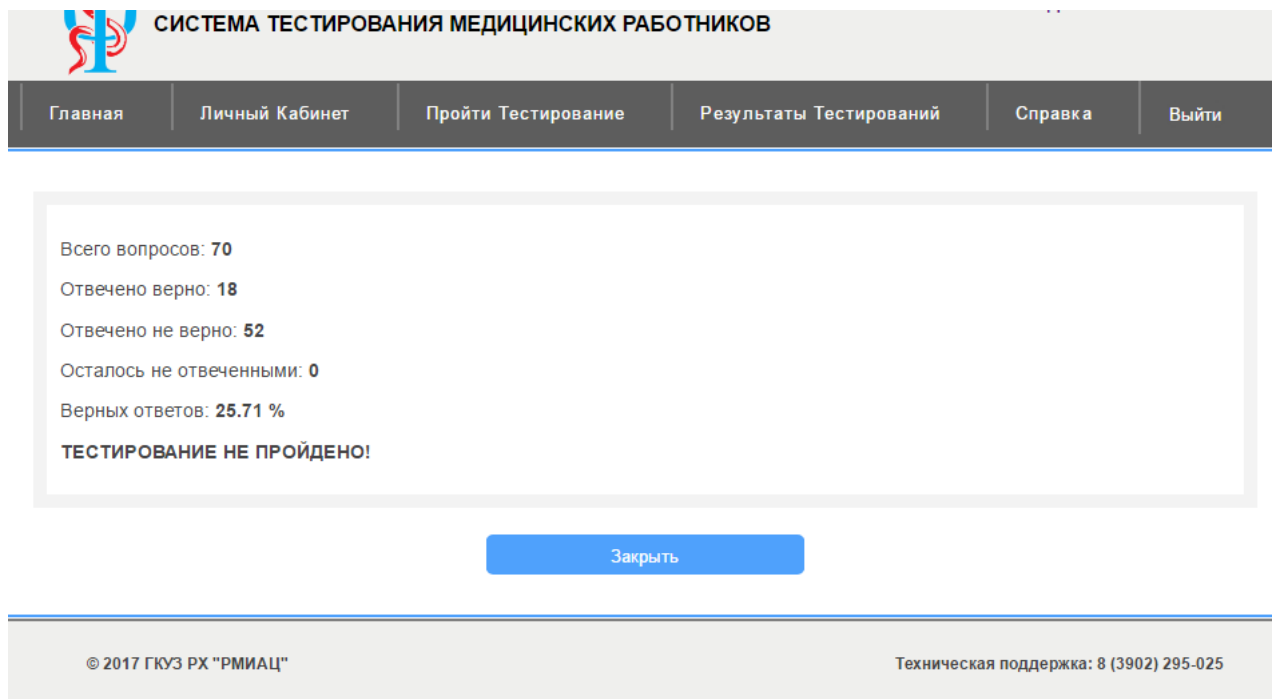


Рисунок 54 – Результат основного тестирования

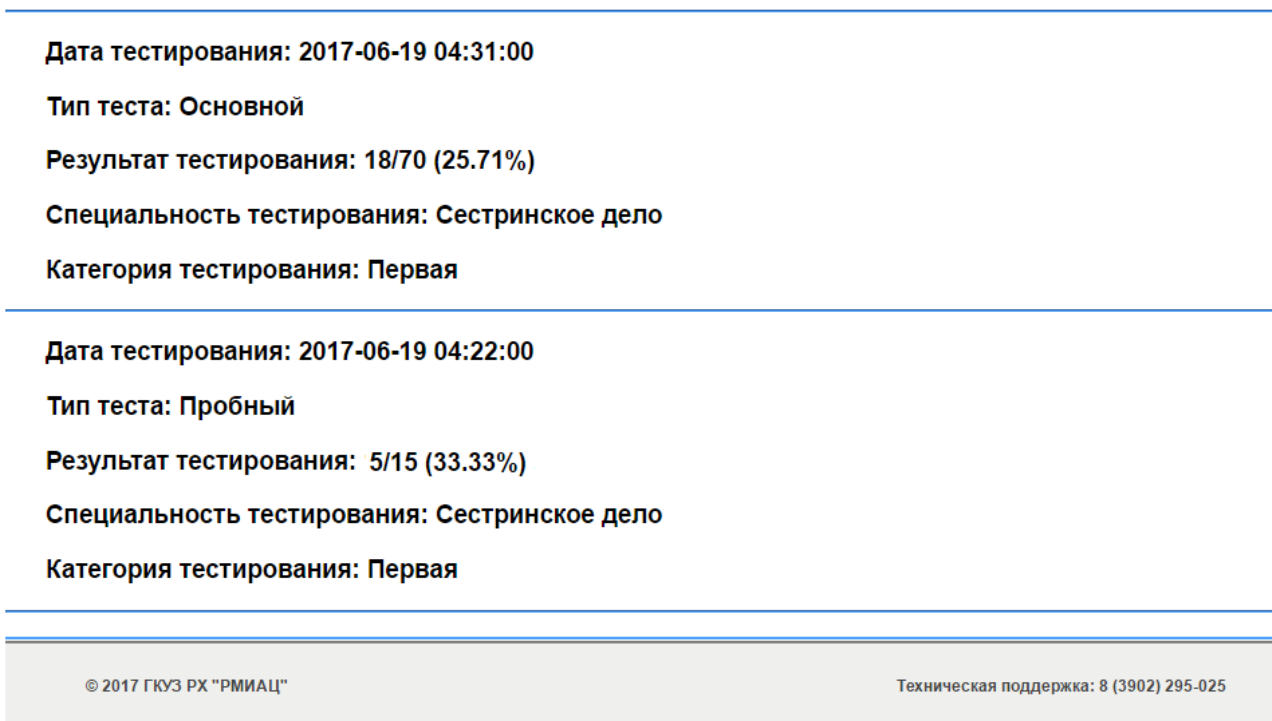



Рисунок 55 – Просмотр результатов тестирований в личном кабинете

СИСТЕМА ТЕСТИРОВАНИЯ МЕДИЦИНСКИХ РАБОТНИКОВ

ДОБРО ПОЖАЛОВАТЬ, МАРИЯ АЛЕКСАНДРОВНА!

[Главная](#) | [Личный Кабинет](#) | [Пройти Тестирование](#) | [Результаты Тестирований](#) | [Справка](#) | [Выйти](#)

Вы использовали основную попытку. Для предоставления дополнительной обратитесь в тех поддержку

Пройти Тестирование

Укажите специальность и категорию для тестирования

Группа специальностей

Специальность

☐ Пробный тест

Начать тестирование

© 2017 ГКУЗ РХ "РМИАЦ" | Техническая поддержка: 8 (3902) 295-025

Рисунок 56 – Ошибка при попытке повторного прохождения теста

## Выводы по проектному разделу

В проектном разделе построены контекстная диаграмма потоков данных (DFD), содержащая процесс тестирования пользователей системы и логическая модель БД, состоящая из накопителей, используемых в системе.

После этого рассмотрена реализация модуля информационной системы, отвечающего за аттестационное тестирование медицинских и фармацевтических работников.

А именно, создание базы данных системы, состоящей из 11 таблиц, реализация функциональных возможностей, необходимых для проведения тестирования пользователей.

Изложены файловая структура, логические модули и страницы системы. Выполнен контрольный пример для апробации и демонстрации разработанной системы тестирования. Описано разграничение прав доступа пользователей (доступ к страницам зависит от статуса авторизации пользователя).

### **3 Расчет экономического эффекта от внедрения информационной системы аттестационного тестирования медицинских работников**

В экономическом разделе рассчитываются единовременные затраты на проектирование системы, стоимость владения разработанной системой; рассматривается оценка внедрения разработанной информационной системы как инвестиционного проекта.

#### **3.1 Расчет капитальных затрат**

Капитальные (единовременные) затраты на ИС носят разовый характер. Свою стоимость они переносят на продукцию по частям за счет амортизационных отчислений. Капитальными их называют потому, что они не утрачиваются, а воспроизводятся.

Затраты на проектирование ( $K_{пр}$ ) рассчитываются по следующей формуле:

$$K_{пр} = K_{зп} + K_{ипс} + K_{свт} + K_{проч} \quad (1)$$

Рассчитаем затраты на заработную плату проектировщику ( $K_{зп}$ ):

Проектирование заняло 2 месяца. При среднем окладе 20 500 рублей затраты на оплату труда разработчика данной задачи составляют:

$$K_{зп} = 20\,500 \text{ руб./мес} \cdot 2 \text{ мес.} = 41\,000 \text{ руб.}$$

Затраты на инструментальные программные средства, необходимые для проектирования комплекса задач равны нулю ( $K_{ипс} = 0$ ), т.к. для проектирования использовались свободно распространяемые программные средства.

Затраты на средства вычислительной техники ( $K_{свт}$ ), необходимой для проектирования комплекса задач:



К затратам на проектирование также необходимо отнести затраты на машинное время в период разработки. Стоимость компьютера составляет 35 000 руб., нормативный срок службы компьютера 5 лет.

Рассчитаем годовой фонд машинного времени: 21 день в месяце · 12 месяцев · 8 часов/день = 2 016 часов.

Рассчитаем стоимость одного часа машинного времени. Для этого определим следующие составляющие:

– Сумму годовых амортизационных отчислений по формуле:

$$A_z = C_b \cdot N_{ам} \quad (2)$$

Таким образом при стоимости компьютера  $C_b = 35\,000$  руб. и норме амортизации  $N_{ам} = 25\%$ , амортизация будет составлять:

$$A_z = 35\,000 \cdot 0,25 = 8\,750 \text{ руб. в год}$$

– Затраты на обслуживание составляют 480 руб. в месяц за 1 ПК. 480 руб. в месяц · 12 месяцев = 5 760 руб. в год.

Следовательно, стоимость 1 часа машинного времени составляет:

$$(8\,750 \text{ руб. в год} + 5\,760 \text{ руб. в год}) / 2\,016 \text{ часов} = 7,2 \text{ руб./час.}$$

На работу с ЭВМ приходится: 2 месяца · 21 день в месяце · 6 часов работы в день = 252 часов.

Затраты на машинное время за весь период разработки составили:

$$K_{свт} = 7,2 \text{ руб./час} \cdot 252 \text{ часа} = 1\,815 \text{ руб.}$$

Прочие затраты на проектирование ( $K_{проч}$ ) равны 3% от суммы общих затрат на проектирование:

$$K_{проч} = (41\,000 + 0 + 1\,815) \cdot 0,03 = 1\,285 \text{ руб.}$$

Общая сумма затрат на проектирование:

$$K_{пр} = 41\,000 \text{ руб.} + 0 \text{ руб.} + 1\,815 \text{ руб.} + 1\,285 \text{ руб.} = 44\,100 \text{ руб.}$$

Расчет затрат на технические средства управления ( $K_{тс}$ ):

Стоимость компьютера составляет 35 000 руб., срок службы равен 4 годам, таким образом, коэффициент износа составляет 80%. Остаточная

стоимость компьютера составит 7 000 руб. На задачу тестирования на компьютере приходится около 10% рабочего времени ЭВМ.

$$K_{тс} = 7\,000 \text{ руб.} \cdot 10\% = 700 \text{ руб.}$$

Затраты на создание линий связи локальных сетей не учитываются, так как на предприятии уже существует локальная сеть, способная выдержать нагрузку внедренной ИС аттестационного тестирования медицинских работников.

$$K_{лс} = 0.$$

Затраты на программные средства также не учитываются, т.к. часть ПО приобретена ранее и не требует обновления, а другая часть является свободно распространяемой.

$$K_{по} = 0.$$

Затраты на формирование информационной базы ( $K_{ио}$ ).

Формирование информационной базы заняло у разработчика 10 дней.

$$20\,500 / 21 = 977 \text{ руб./день} - \text{заработная плата разработчика в день.}$$

10 дней  $\cdot$  977 руб. в день = 9 770 руб. – заработная плата разработчика на формирование информационной базы.

На работу с ЭВМ при создании информационной базы потребовалось:

$$7,2 \text{ руб. в час} \cdot 6 \text{ часов} \cdot 10 \text{ дней} = 432 \text{ руб.}$$

Сумма затрат на формирование информационной базы составляет:

$$K_{ио} = 9\,770 \text{ руб.} + 432 \text{ руб.} = 10\,202 \text{ руб.}$$

Обучение персонала состоит из двух составляющих. Необходимо обучить администратора и модератора системы. Время обучения администратора равно 10 часам, модератора – 7 часам.

Можно рассчитать оклад обучаемого специалиста в час.

$$20\,500 \text{ руб.} / 21 \text{ день} / 6 \text{ часов/день} = 163 \text{ руб./час}$$

Таким образом затраты на обучение персонала ( $K_{об}$ ) будут равны:

$$K_{об} = (163 \text{ руб./час} \cdot 10 \text{ часов}) + (163 \text{ руб./час} \cdot 7 \text{ часов}) = 2\,771 \text{ руб.}$$

Расчет затрат на опытную эксплуатацию ( $K_{оэ}$ ).

Опытная эксплуатация проводилась разработчиком в течение 8 дней.

8 дней · 977 руб. в день = 7 816 руб. – заработная плата разработчика за время опытной эксплуатации.

На работу с ЭВМ потребовалось: 7, 2 руб. в час · 8 дней · 5 часов/день = 288 руб.

Общая сумма затрат на опытную эксплуатацию составляет:

$$K_{\text{оэ}} = 7\,816 \text{ руб.} + 288 \text{ руб.} = 8\,104 \text{ руб.}$$

Общая сумма капитальных затрат будет рассчитана по формуле 3. После чего представим в таблице 15 все статьи затрат.

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{лс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}} \quad (3)$$

$$K = 44\,100 + 700 + 0 + 0 + 10\,202 + 2\,754 + 8\,104 = 65\,860 \text{ руб.}$$

Таблица 15 – Капитальные затраты проектирования ИС

Затраты	Состав затрат	Планируемая сумма
Затраты на проектирование ИС	Затраты на заработную плату проектировщиков;	41 000 руб.
	Затраты на инструментальные программные средства для проектирования;	0 руб.
	Затраты на средства вычислительной техники для проектирования	1 815 руб.
	Прочие затраты на проектирование	1 285 руб.
Затраты на технические средства управления		700 руб.
Затраты на создание линий связи локальных сетей		0 руб.
Затраты на программные средства		0 руб.
Затраты на формирование информационной базы	ЗП разработчика	9 770 руб.
	Работа ЭВМ	432 руб.

## Окончание таблицы 15

Затраты на обучение персонала	ЗП обучаемого	2 754 руб.
Затраты на опытную эксплуатацию	ЗП разработчика	7 816 руб.
	Работа ЭВМ	288 руб.

После подсчета капитальных затрат представим данные таблицы в виде диаграммы (рисунок 57).

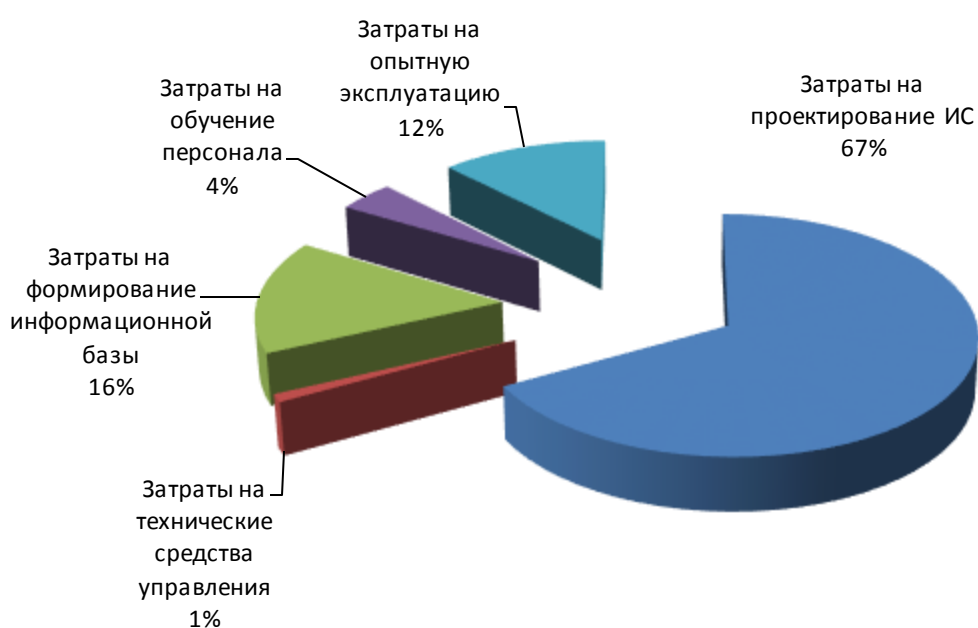


Рисунок 57 – Соотношение статей капитальных затрат по проекту

### 3.2 Расчет совокупной стоимости владения информационной системой

Показатель совокупной стоимости владения ИС состоит из прямых и косвенных расходов и рассчитывается по формуле:

$$TCO = DE + IC \quad (4)$$

При этом прямые расходы (DE) включают в себя следующие группы:

$$DE = DE1 + DE2 + DE3 + DE4 + DE5 + DE6 + DE7 + DE8 \quad (5)$$

*DE1* - капитальные затраты, рассчитаны ранее и равны 62 821 руб.

Расходы на управление ИТ (*DE2*) состоят из 10% от заработной платы администратора системы, так как на работу с системой приходится 10% рабочего времени: 20 500 руб./мес · 12 мес · 0,1 = 24 600 руб./год.

Расходы на техническую поддержку АО и ПО (*DE3*):

Затраты на техническое обслуживание ПК рассчитаны ранее и составляют 5 760 руб. в год. Дополнительно необходимо обслуживать сервер. Затраты на обслуживание одного сервера составляют 700 руб. в месяц. 700 руб. в месяц · 12 месяцев = 8 400 руб. в год. Итого затраты на техническую поддержку ПО будут равны: 5 760 + 8 400 = 14 160 руб. / год.

Затраты на техподдержку ПО будут равны 10% в год, с учетом возможного финансирования доработки системы и прочих затрат:

$$44\,100 \text{ руб} \cdot 0,1 = 4\,410 \text{ руб. в год.}$$

Таким образом,  $DE3 = 14\,160 + 4\,410 = 18\,570$  руб./год.

В процессе владения системой тестирования разработка прикладного ПО не требуется, поэтому затраты на разработку прикладного ПО внутренними силами в расчет не берутся ( $DE4 = 0$ ).

Расходы на аутсорсинг не учитываются, так как все работы будут выполняться внутренними силами предприятия, без привлечения сторонних компаний и специалистов ( $DE5 = 0$ ).

При владении данной системой командировки сотрудников не требуются, так как доступ пользователей к системе осуществляется дистанционно, а администратор системы является сотрудником предприятия владельца ИС ( $DE6 = 0$ ).

К расходам на услуги связи (*DE7*) можно отнести оплату услуг Интернет провайдера: 350 руб./мес. · 4 мес. = 1 400 руб.

Другие группы расходов ( $DE8$ ) рассчитываются как 3% от всех прямых расходов:  $(62\,821 + 24\,600 + 18\,570 + 0 + 0 + 0 + 1\,400) \cdot 0,03 = 3\,222$  руб.

Общая сумма прямых расходов по формуле 5 будет составлять:

$DE = 62\,821 + 24\,600 + 18\,570 + 0 + 0 + 0 + 1\,400 + 3\,222 = 110\,613$  руб. / год.

Косвенные «Непрямые» затраты ( $IC$ ) будут включать отвлечение сотрудников от своих должностных обязанностей в связи с работами загрузки и актуализации рабочей информации, справочной системы, обновлению справочников системы.

Так как обновление справочников осуществляется довольно редко и в небольших объемах, 2 часа в месяц для этого будет достаточно.

$20\,500$  руб. / 21 день / 6 часов в день =  $163$  руб./час – заработная плата сотрудника в час.

$163$  руб. в час  $\cdot$  2 часа в месяц =  $326$  руб./месяц – заработная плата сотрудника в месяц.

Другие статьи косвенных затрат ( $IC$ ) не учитываются или включены в другие группы прямых расходов. Таким образом сумма косвенных затрат будет равна:

$IC = 326$  руб./мес.  $\cdot$  12 мес. =  $3\,912$  руб./год.

Исходя из этого, по формуле 4 рассчитаем совокупную стоимость владения системой:

$TCO = 110\,613 + 3\,912 = 114\,525$  руб./год.

### 3.3 Оценка внедрения ИС как инвестиционного проекта

Так как внедрение ИС аттестационного тестирования медицинских работников не имеет чистого дохода в виде денежных средств, для эффективности расчетов используются качественные методы оценки.

Основными характеристиками, влияющими на качественные методы оценки, будут являться: подтверждение медицинского работника, возможность

управления системой Министерством здравоохранения РХ, автоматизация выгрузки вопросов, свободный авторизованный доступ к результатам тестирования, возможность изменения личных данных.

Для качественной оценки введем показатель повышения качества тестирования и произведем ранжирование характеристик по следующим группам: «значительно повышает качество тестирования», «повышение средней значимости» и «малозначимое повышение качества» (таблица 17).

Таблица 17 – Качественные характеристики

Наименование качественной характеристики	Текущая система тестирования	Внедряемая система тестирования	Показатель повышения качества тестирования
Подтверждение медицинского работника	Отсутствует	Реализована	Значительный
Возможность управления системой Министерством здравоохранения РХ	Отсутствует	Отвечает за наполнение банка вопросов, ведет отчетность	Значительный
Автоматизация выгрузки вопросов	Функционирует с ошибками	Реализована	Значительный
Свободный авторизованный доступ к результатам тестирования	Реализована только для администратора	Реализован	Средней значимости
Возможность изменения личных данных	Отсутствует	Реализована	Малозначимый

Анализируя данные таблицы можно сказать, что разрабатываемая информационная система аттестационного тестирования по качественным показателям превосходит текущую систему.

### **Вывод по экономическому разделу**

Для обоснования экономической эффективности оценены капитальные затраты на разработку системы аттестационного тестирования медицинских работников, которые составили 65 860 рублей. Оценена стоимость совокупного владения информационной системой – 114 525 рублей.

По качественным показателям оценки внедрения ИС (функции подтверждения медицинского работника, автоматизированной выгрузки вопросов; возможность управления системой Министерством здравоохранения РХ; доступ к результатам тестирования и возможность изменения персональных данных в личном кабинете), разработанная система аттестационного тестирования медицинских работников превосходит систему, используемую в настоящее время.



## **ЗАКЛЮЧЕНИЕ**

Выпускная квалификационная работа состоит из трех разделов: аналитического, практического и экономического.

В разделе «Анализ деятельности Министерства Здравоохранения Республики Хакасия в области аттестационного тестирования» произведен анализ деятельности Министерства здравоохранения, процесса аттестации медицинских и фармацевтических работников. Построены диаграммы потоков данных процесса аттестации медицинских и фармацевтических работников.

В ходе анализа используемых программных продуктов выявлено, что используемая система тестирования требует модернизации или полной ее замены.

Проведен сравнительный анализ готовых информационных систем тестирования сотрудников, по результатам которого принято решение о собственной разработке системы.

Произведено сравнение и выбор средств разработки веб-приложений. Основным языком выбран PHP. Также использованы язык гипертекстовой разметки (HTML), каскадные таблицы стилей (CSS) и клиентский язык программирования (JavaScript).

В качестве СУБД использовано средство MySQL, визуальный веб-интерфейс phpMyAdmin.

В разделе «Разработка модуля, реализующего процесс тестирования медицинских работников» рассматривается проектирование и создание базы данных тестирования, файловая структура проекта и описаны функциональные возможности системы, необходимые для проведения тестирования пользователей.

Цель и задачи выпускной квалификационной работы достигнуты. Результатом данной работы является разработанная информационная система аттестационного тестирования медицинских работников.

В разделе «Расчет экономического эффекта от внедрения информационной системы аттестационного тестирования медицинских работников» для обоснования экономической эффективности оценены капитальные затраты на разработку системы аттестационного тестирования медицинских работников, которые составили 65 860 рублей. Оценена стоимость владения информационной системой – 114 525 рублей.

Разработанная система аттестационного тестирования медицинских работников превосходит систему, используемую в настоящее время по таким качественным показателям оценки внедрения информационных систем, как функции подтверждения медицинского работника и автоматизированной выгрузки вопросов; возможность управления системой сотрудником Министерства здравоохранения РХ; личный кабинет пользователя для доступа к результатам тестирований и изменения личных данных.

## СПИСОК СОКРАЩЕНИЙ

В данной выпускной квалификационной работе применены следующие сокращения:

АТ – аттестационное тестирование;

ГКУЗ – государственное казенное учреждение здравоохранения;

ИС – информационная система;

МЗ – министерство здравоохранения;

МО – медицинская организация;

ПДн – персональные данные;

РМИАЦ – республиканский медицинский информационно-аналитический центр.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Постановление Правительства Республики Хакасия от 11 июня 2009 г. N 260 "Об утверждении Положения о Министерстве здравоохранения Республики Хакасия" (с изменениями и дополнениями) [Электронный ресурс] – Режим доступа: <http://base.garant.ru>
2. Постановление Правительства Республики Хакасия от 21 августа 2007 г. N 264 "Об утверждении уставов государственных учреждений здравоохранения Республики Хакасия" [Электронный ресурс] : постановление правительства РХ от 21.08.2007 в ред 26.08.2014. // Справочная правовая система «Гарант». – Режим доступа: <http://base.garant.ru>.
3. Приказ Министерства здравоохранения РФ от 23 апреля 2013 г. N 240н "О Порядке и сроках прохождения медицинскими работниками и фармацевтическими работниками аттестации для получения квалификационной категории" [Электронный ресурс] – Режим доступа: <http://base.garant.ru>
4. Методология IDEF0. Стандарт. Русская версия
5. Ананченко И.В. Классификация компьютерных систем тестирования знаний учащихся // Международный журнал экспериментального образования. – 2016. – № 4-2. – С. 210-213; [Электронный ресурс] – Режим доступа: <https://expeducation.ru>
6. Избачков Ю. С. Информационные системы: Учебник для вузов. 2-е изд. / Ю. С. Избачков, В. Н. Петров. – СПб.: Питер, 2006. – 656 с
7. PHP, Ruby, Python – краткая характеристика трёх языков программирования // Веб-программирование [Электронный ресурс] – Режим доступа: <http://www.internet-technologies.ru>
8. Основы World Wide Web (WWW) // Конспект лекций по веб-программированию [Электронный ресурс] – Режим доступа: <http://www.4stud.info>
9. Диаграмма потоков данных (DFD) [Электронный ресурс] – Режим доступа: <http://e-educ.ru>

10. Веллинг Люк. Разработка Web-приложений с помощью PHP и MySQL, 3-е издание. / Люк Веллинг, Лора Томсон. : Пер. с англ. – М. : Издательский дом «Вильямс», 2008 – 880 с. : ил. – Парал. Тит. Англ.

11. Информационные системы: Учебник для вузов. 3-е изд. / Ю. С. Избачаков, В. Н. Петров, А. А. Васильев, И. С. Телина — СПб.: Питер, 2011. — 544 с.: ил.

## ПРИЛОЖЕНИЕ А

### Программные продукты

Таблица А1 – Программные продукты, имеющиеся в распоряжении органа управления здравоохранением

№ п/п	Наименование программного продукта	Уровень разработки программы	Сфера (область) использования	Краткое описание программного продукта	Пользователи программы
1	РИАМС «ПроМед»	региональный	Деятельность в сфере здравоохранения	Система автоматизации документооборота для лечебно-профилактических учреждений, в которой объединены электронные медицинские карты, данные медицинских исследований, финансовая и административная информация и т.д.	Минздрав Хакасии, медицинские организации
2	Электронная регистратура, Web-сервис врачу19.pф k-vrachu.mz19.ru	региональный	Электронная запись на прием к врачу	Предоставление государственной услуги «Заявка (запись) на прием к врачу» в электронном виде	Минздрав Хакасии, медицинские организации
3	Программный комплекс АСУЛОН	региональный	Льготное лекарственное обеспечение жителей региона	Выписка рецептов на лекарственные средства и изделия медицинского назначения	Минздрав Хакасии, медицинские организации
4	Федеральный регистр медицинских работников	федеральный	Мониторинг кадрового состава системы здравоохранения	Сведения о медицинских работниках Республики Хакасия	Минздрав Хакасии, медицинские организации
5	Федеральный регистр медицинских организаций	федеральный	Мониторинг информации о медицинских организациях	Данные о медицинских организациях и их структурных подразделениях, оказывающих медицинскую помощь	Минздрав Хакасии, медицинские организации

Продолжение таблицы А1

6	Федеральный регистр лиц, инфицированных ВИЧ	федеральный	Мониторинг данные о пациентах, больных ВИЧ	Сбор, обработка и хранение данных пациентов, ведение карт лечения, диспансерного наблюдения, информация об исследованиях, лекарственных препаратах	Минздрав Хакасии, медицинские организации
7	Федеральный регистр лиц, больных туберкулезом	федеральный	Мониторинг данных о пациентах, больных туберкулезом	Сбор, хранение, просмотр и анализ данных о пациентах, больных туберкулезом, просмотр форм отчетности.	Минздрав Хакасии, медицинские организации
8	Федеральный регистр онкологических больных	федеральный	Данные о пациентах, больных онкологическими заболеваниями	Данные о пациентах, больных онкологическими заболеваниями	ГБУЗ РХ «Республиканский клинический онкологический диспансер»
9	Система АХД на базе 1С: Медицина	региональный	Административно-хозяйственная деятельность	Сбор информации об административно-хозяйственной деятельности медицинских организаций	Медицинские организации
10	Федеральный регистр диспансеризации несовершеннолетних декретированных возрастов	федеральный	Учет диспансеризации	Ведение регистра диспансеризации несовершеннолетних декретированных возрастов	Медицинские организации
11	Подсистема мониторинга проведения диспансеризации детей-сирот и детей, находящихся в трудной жизненной ситуации	федеральный	Учет диспансеризации	Диспансеризация детей, находящихся в стационарных учреждениях, детей сирот и детей, оставшихся без попечения родителей	Минздрав Хакасии, медицинские организации

Продолжение таблицы А1

12	Подсистема мониторинга санаторно-курортного лечения	федеральный	Санаторно-курортное лечение	Направление на санаторно-курортное лечение	Минздрав Хакасии
13	Подсистема мониторинга реализации государственного задания по оказанию высокотехнологичной медицинской помощи за счет средств федерального бюджета	федеральный	Высокотехнологичная медицинская помощь	Направление граждан на высокотехнологичную медицинскую помощь	Минздрав Хакасии, медицинские организации
14	Федеральный регистр 7 нозологий	федеральный	Мониторинг сведений о пациентах	Автоматизация процессов сбора, обработки, передачи, хранения, распределённого доступа и поиска лиц, больных гемофилией, муковисцидозом, гипофизарным нанизмом, болезнью Гоше, злокачественными новообразованиями лимфоидной, кроветворной и родственных им тканей, рассеянным склерозом, лиц после трансплантации органов и (или) тканей	Минздрав Хакасии, медицинские организации
15	Подсистема ведения Федерального регистра лиц, страдающих жизнеугрожающим и хроническими прогрессирующими редкими (орфанными) заболеваниями	федеральный	Мониторинг сведений о пациентах	Автоматизация процессов сбора, обработки, передачи, хранения, распределённого доступа и поиска лиц, страдающих жизнеугрожающими и хроническими прогрессирующими редкими (орфанными) заболеваниями	Минздрав Хакасии, медицинские организации



Окончание таблицы А1

16	Автограф	региональный	Диспетчеризация санитарного транспорта на основе ГЛОНАСС	Система мониторинга автомобильного транспорта позволяет видеть точное местоположение машины на карте и измерить простой и пробег, выбрать целесообразный маршрут движения	Медицинские организации
17	ИПРА-Веб	региональный	Ведение карт реабилитации инвалидов	Получение и направление в МСЭ исполненных карт реабилитации инвалидов	Минздрав Хакасии, медицинские организации

## ПРИЛОЖЕНИЕ Б

### Программный код системы

#### config.php

```
<?php
define("HOST", "localhost");
define("USER", "root");
define("PASS", "1510");
define("DB", "testing");
$db = @mysqli_connect(HOST, USER, PASS, DB) or die('Нет соединения с БД');
mysqli_set_charset($db, 'utf8') or die('Не установлена кодировка соединения');
$count_quest = 70; //количество вопросов в тесте
?>
```

#### header.php

```
<script type="text/javascript" src=http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js
></script>
<?php
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
if (!isset($_SESSION['msg'])) {
    $_SESSION['msg'] = ''; }
if (isset($_POST['login']) && isset($_POST['password'])) {
    $msg = login($_POST);
    if ($msg === TRUE) {
        if (($_SESSION['a'] == 1) or ($_SESSION['a'] == 2)) {
            header("Location: adminresult.php"); }
        else { header("Location: kabinet.php");}}
    else {
        $_SESSION['msg'] = $msg;
        header("Location:".$_SERVER['PHP_SELF']); }
    exit(); }
if (isset($_POST['logout'])) {
    $msg = logout();
    if ($msg === TRUE) {
        header("Location:".$_SERVER['PHP_SELF']);
        exit(); }
}
?>
<html>
<head>
<title> TMC </title>
<link rel="shortcut icon" href="images//siteicon.ico" type="image/x-icon">
<meta charset="utf-8">
<link rel="stylesheet" type="text/css" href="styles.css">
```

```

</head>
<body>
<div class="page">
<div class="header">
<div class="logo">
  <a href="index.php"></a>  <h1> Система
тестирования медицинских работников </h1>
</div>
<?php if (!isset($_SESSION['sess'])): // выводим окно входа ?>
<div class="searchform">
  <nav>
    <ul>
      <li id="login">
        <a id="login-trigger" href="#">
          Войти <span>&#x25BC;</span>
        </a>
        <div id="login-content">
          <form method="POST" action="" style="margin: 0px">
            <input id="username" type="text" name="login"
placeholder="Логин" required>
            <input id="password" type="password" name="password"
placeholder="Пароль" required>
            <label><input type="checkbox" name="member" value="1"
style="width: 20px;"> Запомнить меня</label>
            <input type="submit" style="width: 110px;" value="Войти">
          </form>
          <a href="returnpass.php"><label>Забыли пароль?</label></a>
        </div> <!--id="login-content"-->
      </li>
      <li id="signup">
        <a href="register.php">Регистрация</a>
      </li>
    </ul>
  </nav>
</div> <!--class="searchform"-->
<?php else: //выводим окно выхода ?>
  <div class="searchform">
    <div id="login-content">
      <h1>Добро пожаловать, <a href="kabinet.php">
        <?=$_SESSION['name']?> <?=$_SESSION['patronymic']?>!</a></h1>
    </div> <!--id="login-content"-->
  </div> <!--class="searchform"-->
<?php endif;?>
</div> <!--class="header"-->

<form style="margin-bottom: 0px;" method="POST">
<div class="menu-main">
  <ul class="menu-main">

```

```

        <?php if (isset($_SESSION['a']) && (($_SESSION['a'] == 1) || ($_SESSION['a'] == 2)) &&
isset($_SESSION['sess'])):?>
            <li><a href="adminresult.php">Результаты тестирований</a></li>
            <li><a href="adminusers.php">Пользователи</a></li>
            <li><a href="adminquest.php">Банк вопросов</a></li>
        <?php if ($_SESSION['a'] == 1):?>
            <li><a href="adminbook.php">Справочники</a></li>
        <?php endif; ?>
            <li><a href="adminexpected.php">Список ожидаемых</a></li>
            <li><a href="adminreports.php">Отчеты</a></li>
        <?php else: ?>
            <li><a href="index.php">Главная</a></li>
        <?php if (isset($_SESSION['sess'])):?>
            <li><a href="kabinet.php">Личный кабинет</a></li>
            <li><a href="testp.php">Пройти тестирование</a></li>
            <li><a href="kabinetresult.php">Результаты тестирований</a></li>
        <?php endif; ?>
            <li><a href="reference.php">Справка</a></li>
        <?php endif; ?>
        <?php if (isset($_SESSION['sess'])):?>
            <li><a><input type="submit" id="logout" name="logout" value="Выйти"></a></li>
        <?php endif; ?>
    </ul>
</div> <!--class="menu-main"-->
</form>
<div class="errors">
    <?php if (!empty($_SESSION['msg'])): ?>
        <br>
        <?=$_SESSION['msg']?>
        <?php unset($_SESSION['msg'])?>
        <br>
    <?php endif;?>
</div> <!--class="errors"-->
<div class="content">

```

## footer.php

```

</div>
<div class="footer">
    <div class="cop">
        <h2> © 2017 ГКУЗ РХ "РМИАЦ" </h2>
    </div>
    <div class="teh">
        <h2> Техническая поддержка: 8 (3902) 295-025 </h2>
    </div>
</div> <!--class="footer"-->
</div> <!--class="page"-->
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script src="scripts.js"></script>

```

```
</body>
</html>
```

## **index.php**

```
<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
?>
<?php require_once 'header.php'; ?>
<h3>Система предназначена для прохождения аттестационного тестирования
медицинскими работниками. </h3>
<div class="menu-graf">
<div class="menu-gr">
    <ul>Для тестирования необходимо зарегистрироваться или авторизоваться (если уже
    есть учетная запись) в системе.</ul>
    <ul>В СПРАВКЕ описано, как работать с системой пользователю. </ul>
    <ul>Если Вы не можете войти в систему, обратитесь в техническую поддержку.</ul>
</div>
<div class="menu-grafik">
    <a href="https://www.mz19.ru/razdely/kadrovaya_politika/list.php?SECTION _ID=257">
    Посмотреть график заседаний аттестационных комиссий, документацию к аттестации
    ВРАЧЕЙ</a>
</div>
<div class="menu-grafik">
    <a href="https://www.mz19.ru/razdely/kadrovaya_politika/list.php?SECTION _ID=256">
    Посмотреть график заседаний аттестационных комиссий, документацию к аттестации
    СРЕДНЕГО МЕДИЦИНСКОГО ПЕРСОНАЛА</a>
</div>
</div>
<?php require_once 'footer.php'; ?>
```

## **reference.php**

```
<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
?>
<?php require_once 'header.php'; ?>
    <div class="sprafka">
    <h2>Справка </h2>
    <ul>Для тестирования необходимо зарегистрироваться или авторизоваться (если
    уже есть учетная запись) в системе.</ul>
```

## <h2>Регистрация</h2>

<ul>При регистрации необходимо указать логин и пароль, которые будут в дальнейшем использоваться для авторизации в системе; личные данные пользователя, адрес электронной почты, место работы, специальность, стаж и медицинскую категорию.</ul>

<ul>Для удобства выбора, специальности подразделяются на группы, поэтому перед выбором специальности необходимо указать группу, к которой она относится.</ul>

<ol>Нажимая кнопку "Регистрация" вы даете свое согласие на обработку персональных данных:

<li>Я, именуемый(ая) в дальнейшем Субъект, в соответствии с требованиями статьи 9 Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных» подтверждаю свое согласие на обработку ГКУЗ РХ «МИАЦ» (далее – Оператор) моих персональных данных, представляемых для внесения в реестр сведений для регистрации на сайте Оператора, включающих: фамилию, имя, отчество; контактный номер телефона, адрес электронной почты, специальность, место работы, должность, стаж работы.</li>

<li>Предоставляю Оператору право осуществлять все действия (операции) с моими персональными данными, включая сбор, систематизацию, накопление, хранение, обновление, изменение, использование, обезличивание, блокирование, уничтожение. Оператор вправе обрабатывать мои персональные данные посредством внесения их в электронную базу данных, включения в списки (реестры) и отчетные формы, предусмотренные документами, регламентирующими предоставление отчетных данных (документов).</li>

<li>Оператор имеет право на обмен (прием и передачу) моими персональными данными с использованием машинных носителей или по каналам связи, с соблюдением мер, обеспечивающих их защиту от несанкционированного доступа, при условии, что их прием и обработка будут осуществляться лицом, обязанным сохранять профессиональную тайну.</li>

<li>Настоящее согласие действует 5 лет в соответствии перечнем типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения.</li>

<li>В соответствии с п.4 ст. 14 Федерального закона «О персональных данных» от 27.06.2006 г. №152 Субъект персональных данных по письменному запросу имеет право на получение информации, касающейся обработки его персональных данных.</li>

<li>Обработка персональных данных, не включенных в общедоступные источники, прекращается при поступлении Оператору письменного заявления Субъекта о расторжении Согласия.</li>

<li>При поступлении Оператору письменного заявления Субъекта о прекращении действия Согласия, персональные данные деперсонализируются в 15-дневный срок.</li>

</ol>

## <h2>Тестирование </h2>

<ul>Перед началом тестирования необходимо выбрать специальность и медицинскую категорию по которым производится аттестация.</ul>

<ul>

<li>Тест состоит из 70 вопросов.</li>

<li>Ограничение по времени - 95 минут.</li>

<li>Ограничение количества основных попыток - 1.</li>

<li>Обязательно прохождение пробного тестирования.</li>

```

        </ul>
        <h2> Личный кабинет </h2>
        <ul>В личном кабинете отображаются все данные пользователя, которые, при
        необходимости можно изменить.</ul>
        <li>Здесь же хранятся результаты всех тестирований пользователя.</li>
        <h2> Рекомендации </h2>
        <ul>Рекомендуется использовать браузеры последних версий, такие как:
        <li> Google Chrome </li>
        <li>Firefox </li>
        <li>Internet Explorer (версии 9) </li>
        </ul>
        <h2>Техническая Поддержка </h2>
        <table>
            <tr>
                <td> Контактные телефоны</td>
                <td>Техническая поддержка: 8 (3902) 295-025 </td>
            </tr>
            <tr>
                <td> Электронная почта</td>
                <td> medstat@r-19.ru</td>
            </tr>
            <tr>
                <td> Часы работы техподдержки </td>
                <td> 8-00 до 16-30 </td>
            </tr>
        </table>
    </div>
    <?php require_once 'footer.php'; ?>

```

## register.php

```

<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
$groups = get_group();
$specialty = get_spec();
$facility = get_facility();
$category = get_category();
if (!isset($_SESSION['msg'])) {
    $_SESSION['msg'] = '';
}
if (!empty($_POST['code'])) {
    echo getSpecial();
    exit;
}
if (!isset($_SESSION['reg'])) {

```

```

$_SESSION['reg']['login'] = "";
$_SESSION['reg']['email'] = "";
$_SESSION['reg']['surname'] = "";
$_SESSION['reg']['name'] = "";
$_SESSION['reg']['patronymic'] = "";
$_SESSION['reg']['group'] = "";
$_SESSION['reg']['specialty'] = "";
$_SESSION['reg']['experience'] = "";
$_SESSION['reg']['mf'] = "";
$_SESSION['reg']['category'] = "";
$_SESSION['reg']['phone'] = "";
}
if(isset($_POST['reg'])) {
    $msg = registration($_POST);
    if($msg === TRUE) {
        $_SESSION['msg'] = "Вы успешно зарегистрировались на сайте. Для подтверждения
регистрации Вам на почту отправлено письмо с инструкциями.";
    }
    else {
        $_SESSION['msg'] = $msg;
    }
} ?>
<?php require_once 'header.php'; ?>
<h2> Регистрация пользователя</h2>
<form method='POST' id='inpu">
    <label>Логин</label>
    <input type='text' name='reg_login' placeholder='Логин' value="<?=$_SESSION['reg']['login'];?>"
required>
    <br>
    <label>Пароль</label>
    <input type='password' name='reg_password' placeholder='Пароль' required>
    <br>
    <label>Подтверждение пароля</label>
    <input type='password' name='reg_conf_password' placeholder='Повторите пароль' required>
    <br>
    <label>Электронная почта</label>
    <input type='email' name='reg_email' placeholder='Электронная почта'
value="<?=$_SESSION['reg']['email'];?>" required>
    <br>
    <label>Фамилия</label>
    <input type='text' name='reg_surname' placeholder='Фамилия'
value="<?=$_SESSION['reg']['surname'];?>" required>
    <br>
    <label>Имя</label>
    <input type='text' name='reg_name' placeholder='Имя' value="<?=$_SESSION['reg']['name'];?>"
required>
    <label>Отчество</label>
    <input type='text' name='reg_patronymic' placeholder='Отчество'
value="<?=$_SESSION['reg']['patronymic'];?>" required>
    <label>Группа специальностей</label>

```



```

<select class="facility" name="reg_group" required id="group">
  <option disabled selected value="0">Выберите группу</option>
  <?php foreach ($groups as $group): ?>
    <?php if ($group['id'] == $_SESSION['reg']['group']) : ?>
      <option selected value="<?=$group['id']?>"><?=$group['group']?></option>
    <?php else : ?>
      <option value="<?=$group['id']?>"><?=$group['group']?></option>
    <?php endif; ?>
  <?php endforeach; ?>
</select>
<?php if($_SESSION['reg']['group']) : ?>
<label>Специальность</label>
<select class="form-control" name="reg_specialty" required id="special">
  <option disabled selected value="0">Выберите специальность</option>
  <?php foreach ($specialty as $spec): ?>
    <?php if ($spec['id'] == $_SESSION['reg']['specialty']) : ?>
      <option selected value="<?=$spec['id']?>"><?=$spec['specialty']?></option>
    <?php else : ?>
      <option value="<?=$spec['id']?>"><?=$spec['specialty']?></option>
    <?php endif; ?>
  <?php endforeach; ?>
</select>
<?php else : ?>
<label>Специальность</label>
<select class="form-control" name="reg_specialty" id="special" required>
  <option disabled selected value="0">Выберите специальность</option></select>
<?php endif; ?>
<br>
<label>Стаж работы по специальности</label>
<input type="text" name="reg_experience" placeholder="Стаж"
value="<?=$_SESSION['reg']['experience'];?>" required>
<br>
<label>Место работы</label>
<select class="facility" name="reg_mf" required>
  <option disabled selected value="0">Выберите место работы</option>
  <?php foreach ($facility as $m): ?>
    <?php if ($m['id'] == $_SESSION['reg']['mf']) : ?>
      <option selected value="<?=$m['id']?>"><?=$m['med_facility']?></option>
    <?php else : ?>
      <option value="<?=$m['id']?>"><?=$m['med_facility']?></option>
    <?php endif; ?>
  <?php endforeach; ?>
</select>
<br>
<label>Категория</label>
<select class="category" name="reg_category" required>
  <option disabled selected value="0">Выберите категорию</option>
  <?php foreach ($category as $category): ?>
    <?php if ($category['id'] == $_SESSION['reg']['category']) : ?>
      <option selected value="<?=$category['id']?>"><?=$category['category']?></option>
    <?php else : ?>
      <option value="<?=$category['id']?>"><?=$category['category']?></option>
    <?php endif; ?>
  <?php endforeach; ?>
</select>

```

```

        <?php endif; ?>
    <?php endforeach; ?>
</select>
<br>
<label>Телефон</label> +7 <input type='te' maxlength="10" name='reg_phone'
value="<?=$_SESSION['reg']['phone'];?>" placeholder='Телефон' required>
<br>
<label><input type="checkbox" name="reg_consent" value="1" style="width: auto;" required>Я
даю свое согласие на автоматизированную обработку предоставляемых персональных
данных на основании Федерального закона от 27.07.2006 № 152-ФЗ «О персональных
данных»</label>
<input type="submit" name='reg' value="Регистрация">
</form>
<?php unset($_SESSION['reg']); ?>
<?php require_once 'footer.php'; ?>

```

## confirm.php

```

<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
$error = "";
$confirm = "";
if ($_GET['hash']) {
    $confirm = confirm();
    if ($confirm === TRUE) {
        $confirm = "Ваша учетная запись активирована. Можете авторизоваться на
сайте.";
    }
    $confirm .= '<a href="kabinet.php"><button class="center btn"
id="exit">Закрыть</button></a>';
}
else {
    $error = "Неверная ссылка.";
}
?>
<?php require_once 'header.php'; ?>
<h2>Подтверждение учетной записи</h2>
<div id="inp">
    <h2><?=$error;?></h2>
    <h2><?=$confirm;?></h2>
</div>
<?php require_once 'footer.php'; ?>

```

## returnpass.php

```

<?php
session_start();
ini_set("display_errors", 1);

```

```

error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
if (isset($_POST['email'])) {
    $msg = get_password($_POST['email']);
    if ($msg === TRUE) {
        $_SESSION['msg'] = "Новый пароль выслан вам на почту";
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}
?>
<?php require_once 'header.php'; ?>
<h2>Восстановление пароля</h2>
<form method="POST" id="inp">
    <label>Введите адрес почтового ящика для восстановления пароля</label>
    <input type="email" name="email" style="width: 400px" placeholder="e-mail">
    <input type="submit" name="returnpass" value="Восстановить">
</form>
<?php require_once 'footer.php'; ?>

```

## kabinet.php

```

<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
if (!check_user()) {
    header("Location: index.php");
    exit();
}
else {
    if (($_SESSION['a'] == 1) or ($_SESSION['a'] == 2)) {
        header("Location: adminusers.php");
    }
}
$user = get_user($_SESSION);
$facility = get_facility();
foreach ($facility as $key => $value) {
    if ($value['id'] == $user['mf']) {
        $user['mf'] = $value['med_facility'];
    }
}
$category = get_category();
foreach ($category as $key => $value) {
    if ($value['id'] == $user['category']) {

```

```

        $user['category'] = $value['category'];
    }
}
$spec = get_spec();
foreach ($spec as $key => $value) {
    if ($value['id'] == $user['specialty']) {
        $user['specialty'] = $value['specialty'];
    }
}

if (isset($_POST['change'])) {
    header("Location: kabinetchange.php");
}
?>
<?php require_once 'header.php'; ?>
    <h2>Личный кабинет</h2>
    <br>
    <br>
    <h3><?=$user['surname'];?> <?=$user['name'];?> <?=$user['patronymic'];?></h3>
    <br>
    <h3>Место работы: <?=$user['mf'];?></h3>
    <h3>Специальность: <?=$user['specialty'];?></h3>
    <h3>Стаж: <?=$user['experience'];?></h3>
    <h3>Врачебная категория: <?=$user['category'];?></h3>
    <br>
    <h3>Электронная почта: <?=$user['email'];?></h3>
    <h3>Телефон: <?=$user['phone'];?></h3>
    <br>
    <form method="POST">
        <input type="submit" style="margin-left: 32px;" name='change' value="Изменить">
    </form>
    <?php require_once 'footer.php'; ?>

```

## testp.php

```

<?php
session_start();
require_once 'config.php';
require_once 'functions.php';
if (!check_user()) {
    header("Location: index.php");
    exit();
}
$groups = get_group();
$category = get_category();
$_SESSION['id_category'] = '1';
status($_SESSION['user_id']);
if (isset($_POST['start'])) {
    if (isset($_POST['special'])) {

```

```

    $_SESSION['id_spec'] = $_POST['special'];
}
else {
    $_SESSION['msg'] = "Не выбрана специальность";
    header("Location:".$_SERVER['PHP_SELF']); exit();
}

if (isset($_POST['prob']) && ($_SESSION['probe_test'] == 1)) {
    $_SESSION['msg'] = "Вы уже прошли пробный тест";
    header("Location:".$_SERVER['PHP_SELF']); exit();
}
if (!isset($_POST['prob']) && ($_SESSION['probe_test'] == 0)) {
    $_SESSION['msg'] = "Вы не прошли пробный тест";
    header("Location:".$_SERVER['PHP_SELF']); exit();
}
if (!isset($_POST['prob']) && ($_SESSION['main_test'] == 1) && ($_SESSION['main_test1'] == -1))
{
    $_SESSION['msg'] = "Вы использовали основную попытку. Для предоставления
дополнительной обратитесь в тех поддержку";
    header("Location:".$_SERVER['PHP_SELF']); exit();
}
if (!isset($_POST['prob']) && ($_SESSION['main_test'] == 1) && ($_SESSION['main_test1'] == 1)) {
    $_SESSION['msg'] = "Вы использовали все попытки";
    header("Location:".$_SERVER['PHP_SELF']); exit();
}
if (isset($_POST['prob'])) {
    $_SESSION['type_test'] = 'probe';
}
else {
    if ((($_SESSION['main_test'] == 1) && ($_SESSION['main_test1'] == 0))){
        $_SESSION['type_test'] = 'additional';
    }
    else {
        $_SESSION['type_test'] = 'main';
    }
}
header("Location: test.php");
}??>
<?php require_once 'header.php'; ?>
<h2>Пройти Тестирование</h2>
<form method="POST">
<div id="inpu">
    <h2>Укажите специальность и категорию для тестирования</h2>
    <label>Группа специальностей</label>
    <select class="form-control" name="group" id="group">
        <option disabled selected>Выберите группу</option>
        <?php foreach($groups as $group): ?>
            <option value="<?=$group['id']?>"><?=$group['group']?></option>
        <?php endforeach; ?>

```

```

</select>
<label>Специальность</label>
<select class="form-control" name="special" id="special">
    <option disabled selected>Выберите специальность</option>
</select>
<input type="checkbox" name="prob" style="width: auto;" value="1"><label>Пробный
тест</label>
<input type="submit" class="center btn" name="start" value="Начать тестирование">
</div>
</form>
<?php require_once 'footer.php'; ?>

```

## test.php

```

<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
if (!check_user()) {
    header("Location: index.php");
    exit();
}
$id_spec = $_SESSION['id_spec'];
$type_test = $_SESSION['type_test'];
$user_id = $_SESSION['user_id'];
$test_data = get_test_data($id_spec);
$test_id = $_SESSION['test_id'] + 1;
if (is_array($test_data)) {
    $count_questions = count($test_data);
    $pagination = pagination($count_questions, $test_data);
}
if (isset($_POST['spec'])) {
    $test_all_data = get_all_test_data($id_spec);
    $test_result = get_test_result($test_all_data, $test_data, $_POST); //1 - массив всех
вопросов/ответов, 2 - ответы пользователя
    echo print_result($test_result, $test_id, $user_id);
    $inser = insert_result($test_result, $type_test, $test_id, $user_id);
    die;
}
}
<?php require_once 'header.php'; ?>
<div id="timer_wrap">
    <div id="clock">
    </div>
</div>
<div class="wrap">
<div class="contents">
    <h3>Вопросы выбранной специальности</h3>

```

```

<?php if (isset($test_data)) :?>
    <?php if (is_array($test_data)) : ?>
        <p>Всего вопросов: <?=$count_questions?></p>
        <?=$pagination?>
        <span class="none" id="id-spec"><?=$id_spec?></span>
<div class="test-data" id="test-data">
    <?php foreach ($test_data as $id_question => $item): //получаем вопрос + ответы ?>
        <div class="question" data-id="<?=$id_question?>" id="question-<?=$id_question?>">
            <?php foreach ($item as $id_answer => $answer): //перебор массива
                вопрос/ответы?>
                    <?php if (!$id_answer): // выводим вопрос ?>
                        <p class="q"><?=$answer?></p>
                    <?php else: //выводим варианты ответов ?>
                        <p class="a-<?=$id_answer?>">
                            <label id="answer-<?=$id_answer?>-<?=$id_question?>"><input
type="radio" style="width: auto;" id="answer-<?=$id_answer?>" name="question-
<?=$id_question?>" value="<?=$id_answer?>"><?=$answer?></label>
                            </p>
                        <?php endif; // $id_answer ?>
                    <?php endforeach // $item ?>
                </div> <!--question-->
            <?php endforeach; // $test_data ?>
        </div> <!-- .test-data-->
    <div class="buttons">
        <button class="center btn" id="btn">Закончить тест</button>
    </div> <!--.buttons-->
    <?php else: //is_array($test_data)?>
        <h3>Тест в разработке</h3>
    <?php endif; //is_array($test_data)?>
    <?php else: //isset($test_data)?>
        <h3>Выберите тест</h3>
    <?php endif; //isset($test_data)?>
</div> <!-- .content-->
</div> <!-- .wrap -->
<script type="text/javascript" src="scriptsss.js"></script>
<?php require_once 'footer.php'; ?>

```

## kabinetresult.php

```

<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
if (!check_user()) {
    header("Location: index.php");
    exit();
}

```

```

else {
    if (($_SESSION['a'] == 1) or ($_SESSION['a'] == 2)) {
        header("Location: adminusers.php");
    }
}
$spec = get_spec();
$stab = all_result();
$categ = get_category();
foreach ($stab as $k => $v) {
    foreach ($spec as $s) {
        if ($s['id'] == $v['specialty_t']) {
            $stab[$k]['specialty_t'] = $s['specialty'];
        }
    }
    foreach ($categ as $category) {
        if ($category['id'] == $v['category_t']) {
            $stab[$k]['category_t'] = $category['category'];
        }
    }
    if ($v['type_test'] == 'main'){
        $stab[$k]['type_test'] = 'Основной';
    }
    elseif ($v['type_test'] == 'probe') {
        $stab[$k]['type_test'] = 'Пробный';
    }
    elseif ($v['type_test'] == 'additional') {
        $stab[$k]['type_test'] = 'Дополнительная попытка';
    }
    else {
        print("\n\n");
    }
}
?>
<?php require_once 'header.php'; ?>
<h2>Результаты тестирований пользователя</h2>
<br>
<?php foreach ($stab as $key => $value): ?>
    <?php if ($value['user_id'] == $_SESSION['user_id']): ?>
        <h3>Дата тестирования: <?=$value['date']?></h3>
        <h3>Тип теста: <?=$value['type_test']?></h3>
        <h3>Результат тестирования: <?=$value['result']?> (<?=$value['percent']?>%)</h3>
        <h3>Специальность тестирования: <?=$value['specialty_t']?></h3>
        <h3>Категория тестирования: <?=$value['category_t']?></h3>
    <hr>
<?php endforeach; ?>
<?php require_once 'footer.php'; ?>

```

## kabinetchange.php



```

<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
if (!check_user()) {
    header("Location: index.php");
    exit();
}
else {
    if (($_SESSION['a'] == 1) or ($_SESSION['a'] == 2)) {
        header("Location: adminusers.php");
    }
}
$user = get_user($_SESSION);
$facility = get_facility();
$category = get_category();
$specialty = get_spec();
if (isset($_POST['changeuser'])) {
    $msg = change_users($_POST);
    if ($msg === TRUE) {
        $_SESSION['msg'] = "Данные пользователя изменены";
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}
?>
<?php require_once 'header.php'; ?>
    <h2>Изменить личные данные</h2>
    <form method='POST' id="inpu">
        <label>Электронная почта</label>
        <input type='email' name='email' placeholder='Электронная почта'
value="<?=$user['email'];?>" required>
        <br>
        <label>Пароль</label>
        <input type='password' name='password' placeholder='Пароль' required>
        <br>
        <label>Подтверждение пароля</label>
        <input type='password' name='conf_password' placeholder='Повторите пароль' required>
        <br>
        <label>Фамилия</label>
        <input type='text' name='surname' placeholder='Фамилия' value="<?=$user['surname'];?>"
required>
        <br>
        <label>Имя</label>
        <input type='text' name='name' placeholder='Имя' value="<?=$user['name'];?>" required>
        <br>

```

```

<label>Отчество</label>
<input type='text' name='patronymic' placeholder='Отчество'
value="<?=$user['patronymic'];?>" required>
<br>
<label>Специальность</label>
<select class="form-control" name="specialty" required id="special">
  <option disabled selected value="0">Выберите специальность</option>
  <?php foreach ($specialty as $spec): ?>
    <?php if ($spec['id'] == $user['specialty']) : ?>
      <option selected value="<?=$spec['id'];?>"><?=$spec['specialty'];?></option>
    <?php else : ?>
      <option value="<?=$spec['id'];?>"><?=$spec['specialty'];?></option>
    <?php endif; ?>
  <?php endforeach; ?>
</select>
<br>
<label>Стаж работы по специальности</label>
<input type='text' name='experience' placeholder='Стаж' value="<?=$user['experience'];?>"
required>
<br>
<label>Место работы</label>
<select class="facility" name="mf" required>
  <option disabled selected value="0">Выберите место работы</option>
  <?php foreach ($facility as $m): ?>
    <?php if ($m['id'] == $user['mf']) : ?>
      <option selected value="<?=$m['id'];?>"><?=$m['med_facility'];?></option>
    <?php else : ?>
      <option value="<?=$m['id'];?>"><?=$m['med_facility'];?></option>
    <?php endif; ?>
  <?php endforeach; ?>
</select>
<br>
<label>Категория</label>
<select class="category" name="category" required>
  <option disabled selected value="0">Выберите категорию</option>
  <?php foreach ($category as $category): ?>
    <?php if ($category['id'] == $user['category']) : ?>
      <option selected value="<?=$category['id'];?>"><?=$category['category'];?></option>
    <?php else : ?>
      <option value="<?=$category['id'];?>"><?=$category['category'];?></option>
    <?php endif; ?>
  <?php endforeach; ?>
</select>
<br>
<label>Телефон</label> +7 <input type='text' maxlength="10" name='phone'
value="<?=$user['phone'];?>" placeholder='Телефон' required>
<br>
<input type="submit" name='changeuser' value="Сохранить">
<br>

```

```
</form>
<?php require_once 'footer.php'; ?>
```

## scripts.js

```
$(document).ready(function(){
    var gq = new Date();
    var grt = Date.parse(gq) + 95*60*1000; //задаем продолжительность тестирования
    function get_timer() {
        var date_t = new Date(grt);
        var date = new Date();
        var timer = date_t - date;
        if (date_t > date) {
            var min = parseInt(timer / (1000*60));
            if (min<10) {
                min = '0' + min;
            }
            var sec = parseInt(timer / 1000)%60;
            if (sec<10) {
                sec = '0' + sec;
            }
            $("#timer_wrap").html ("<span id='stop'>" + min + ":" + sec + "</span>");
            tim = setTimeout(get_timer, 1000);
        }
        else {
            alert("Время вышло!");
            var spec = +$('#id-spec').text();
            var res = {spec:spec};
            $('.question').each(function(){
                var id = $(this).data('id');
                var user_answer = $('input[name=question-' + id + ']:checked').val();
                var parent_question = $('label[id=answer-parent_question-' + id + ']').text();
                var t = {'user_answer': user_answer,'parent_quest': parent_question};
                res[id] = t;
            });
            console.log(res);
            var ress = JSON.stringify(res);
            console.log(ress);
            $.ajax({
                url:'test.php',
                type: 'POST',
                data: res,
                success: function(html){
                    $('#contents').html(html);
                },
                error: function(){
                    alert('Error!');
                }
            });
        }
    }
});
```

```

    }
  }
  get_timer();
  $('test-data').find('div:first').show();
  $('.pagination a').on('click', function(){
    if ($(this).attr('class') == 'nav-active') return false;
    var link = $(this).attr('href'); //ссылка на текст запрошенной вкладки
    var prevActive = $('.pagination > a.nav-active').attr('href'); //ссылка на текст
    активной вкладки
    $('.pagination > a.nav-active').addClass('nav-visited');
    $('.pagination > a.nav-active').removeClass('nav-active'); //удаляем класс активной
    вкладки
    $(this).addClass('nav-active'); // добавляем класс активной вкладки
    //скрываем и показываем вопросы
    $(prevActive).fadeOut(100, function(){
      $(link).fadeIn(100);
    });
    return false;
  });
  $('#btn').click(function(){
    var spec = $('#id-spec').text();
    var res = {spec:spec};
    $('.question').each(function(){
      var id = $(this).data('id');
      var user_answer = $('input[name=question-' + id + ']:checked').val();
      var parent_question = $('label[id=answer-parent_question-' + id + ']').text();
      var t = {user_answer: user_answer, parent_quest: parent_question};
      res[id] = t;
    });
    console.log(res);
    var ress = JSON.stringify(res);
    console.log(ress);
    $.ajax({
      url:'test.php',
      type: 'POST',
      data: res,
      success: function(html){
        $('.contents').html(html);
      },
      error: function(){
        alert('Error!');
      }
    });
    clearTimeout(tim);
  });
  $('#group').change(function(){
    var code = $(this).val();
    $('#special').load('register.php', {code: code});
  });

```

```

    $('#login-trigger').click(function(){
    $(this).next('#login-content').slideToggle();
    $(this).toggleClass('active');
    if ($(this).hasClass('active')) $(this).find('span').html('&#x25B2;')
    else $(this).find('span').html('&#x25BC;')
    })
});

```

## styles.css

```

body {
margin:0;
padding:0px;
font-size:1em;
font-family: Arial, Helvetica, sans-serif;
color: #444;
margin: 0;
font-size: 16px;
}

```

```

a{
text-decoration: none;
}

```

```

hr {
border-color: #4fa1fc
}

```

```

.page {
width:90%;
margin:-40px auto;
/* padding-bottom: 4em; */
}

```

```

.header {
width:100%;
height:auto;
display:block;
float:left;
background-color:#efefed;
}

```

```

.header h1 {
float:left;
color: black;
font-size:1em;
text-transform:uppercase;
margin-top: 6%;
/*margin: 50px;*/
}

```

```

}

.logotip {
  width: 80px;
  height: 80px;
  float: left;
  margin-top: 2%;
}

.errors {
  width: 100%;
  margin: 0px;
  text-align: center;
  font-size: 1.5em;
  color: #ec430d;
  word-wrap: break-word;
}

.content {
  overflow: hidden;
  width: 100%;
  background-color: white;
  margin: 1px 0px;
  border-bottom: 2px solid #4fa1fc;
  border-top: 2px solid #4fa1fc;
  float: left;
  padding: 10px 0px;
}

.content h2 {
  font-size: 1em;
}

.content p {
  font-size: 0.9em;
}

.footer {
  margin: 0;
  float: left;
  background-color: #efefed;
  border-top: 2px solid gray;
  flex: 0 0 auto;
  width: 100%;
}

.footer h2 {
  padding: 15px;
  font-size: 12px;
}

```

```

}

.cop{
  float: left;
  padding-left: 15px;
}

.teh{
  float: right;
  padding-right: 30px;
}

.logo{
  width: 75%;
  margin: 0 0 50px 0;
  padding-left: 30px;
}

.searchform {
  width: 25%;
  float:right;
  height: 23px;
  font-size:13px;
}

#logout {
  width: 42px;
  height: 20px;
  border: none;
  background: #5d5d5d;
  margin: 0;
  color: white;
  padding: 0px;
}

.menu-main {
  list-style: none;
  margin: 40px 0 0px;
  padding: 25px 0 5px;
  text-align: left;
  background: #5d5d5d;
}

.menu-main li {
  display: inline-block;
  border-left: 2px solid gray;
  margin: 0 0 0 10px;
  color: green;
}

```

```

.menu-main li:after{
  position: absolute;
  color: white;
  display: inline-block;
  vertical-align:top;
  margin: 12px 0 15px 5px;
}

.menu-main li:last-child:after{
  content: none;
}

.menu-main a {
  text-decoration: none;
  letter-spacing: 1px;
  position: relative;
  padding-bottom: 5px;
  margin: 15px 20px 0 20px;
  font-size: 13px;
  text-transform: capitalize;
  display: inline-block;
  /* transition: color .s; */
}

.menu-main a:visited{
  color: white;
}

.menu-main a.current,
.menu-main a:hover{
  color: #4fa1fc;
}

.menu-main a:before,
.menu-main a:after {
  content: "";
  position: absolute;
  height: 4px;
  top: auto;
  right: 50%;
  bottom: -5px;
  left: 50%;
  background: #4fa1fc;
}

.menu-main a:hover:before,
.menu-main .current:before {
  left: 0;
}

```



```

}

.menu-main a:hover:after,
.menu-main .current:after {
    right: 0;
}

@media (max-width: 550px) {
    .menu-main {padding-top: 0;}
    .menu-main li {display: block;}
    .menu-main li:after {content: none; }
    .menu-main a {
        padding: 25px 0 20px;
        margin: 0 30px;
    }
}

input, select {
    padding: 6px 5px;
    margin: 0 0 5px 0;
    width: 238px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

input[type="submit"] {
    display: inline-block;
    color: white;
    text-decoration: none;
    height: 30px;
    padding: .5em 2em;
    border-width: 2px 0;
    border-style: solid none;
    border-color: #4fa1fc #000 #4fa1fc;
    border-radius: 6px;
    background: linear-gradient(#4fa1fc, #4fa1fc) #4fa1fc;
    transition: 0.2s;
    width: 238px;
}

input[type="submit"]:hover {
    background: #a2ceff linear-gradient(#a2ceff, #6ca9ec);
    border-color: #a2ceff #000 #6ca9ec;
}

a.input:hover,
a.input:focus
{
    background: linear-gradient(#79b4f7, 378ff2) #f59500;

```

```

}

fieldset
{
    margin: 0;
    padding: 0;
    border: 0;
}

nav ul {
    margin: 6%;
    padding: 0;
    list-style: none;
    position: relative;
    float: right;
    background: #eee;
    border-bottom: 1px solid gray;
    -moz-border-radius: 3px;
    -webkit-border-radius: 3px;
    border-radius: 3px;
}

    nav li {
        float: left;
    }
nav #login {
    border-right: 1px solid #ddd;
    -moz-box-shadow: 1px 0 0;
    -webkit-box-shadow: 1px 0 0;
    box-shadow: 1px 0 0;
}

nav #login-trigger,
nav #signup a {
    display: inline-block;
    *display: inline;
    *zoom: 1;
    height: 25px;
    line-height: 25px;
    font-weight: bold;
    padding: 0 8px;
    text-decoration: none;
    color: white;
    /* text-shadow: 0 1px 0 #fff; */
}

nav #signup a {
    -moz-border-radius: 0 3px 3px 0;
    -webkit-border-radius: 0 3px 3px 0;

```

```

    border-radius: 0 3px 3px 0;
background: gray;
}

```

```

nav #login-trigger {
    -moz-border-radius: 3px 0 0 3px;
    -webkit-border-radius: 3px 0 0 3px;
    border-radius: 3px 0 0 3px;
background: gray;
}

```

```

nav #login-trigger:hover,
nav #login .active,
nav #signup a:hover {
background: white;
color: gray;
}

```

```

nav #login-content {
    display: none;
position: absolute;
top: 24px;
right: 0;
z-index: 999;
background: white;
padding: 15px;
-moz-box-shadow: 0 2px 2px -1px rgba(0,0,0,.9);
-webkit-box-shadow: 0 2px 2px -1px rgba(0,0,0,.9);
box-shadow: 0 2px 2px -1px rgba(0,0,0,.9);
-moz-border-radius: 3px 0 3px 3px;
-webkit-border-radius: 3px 0 3px 3px;
border-radius: 3px 0 3px 3px;
float: right;
}

```

```

nav li #login-content {
    right: 0;
width: 250px;
}

```

```

#inputs input, #inputs select {
    background: #f1f1f1;
padding: 6px 5px;
margin: 0 0 5px 0;
width: 238px;
border: 1px solid #ccc ;
-moz-border-radius: 3px;
-webkit-border-radius: 3px;
border-radius: 3px;
}

```

```

-moz-box-shadow: 0 1px 1px #ccc inset;
-webkit-box-shadow: 0 1px 1px #ccc inset;
box-shadow: 0 1px 1px #ccc inset;
}

#inputs input:focus, #inputs select:focus {
  background-color: #fff;
  border-color: #4fa1fc;
  outline: none;
  -moz-box-shadow: 0 0 0 1px #4fa1fc inset;
  -webkit-box-shadow: 0 0 0 1px #4fa1fc inset;
  box-shadow: 0 0 0 1px #4fa1fc inset;
}

#login #actions {
  margin: 10px 0 0 0;
}

#login label {
  float: right;
  line-height: 30px;
}

#login label input {
  position: relative;
  top: 2px;
  right: 2px;
}

#inp {clear:both;
  text-align:center;
}

#inpu {clear:both;
  text-align:right;
  float:left;
  width: 70%;}

#inpu input, #inpu select, #inpu span {
  padding: 6px 5px;
  margin: 15px 5 5px 5;
  width: 238px;
  border: 1px solid #4fa1fc;
  -moz-border-radius: 3px;
  -webkit-border-radius: 3px;
  border-radius: 3px;
  -moz-box-shadow: 0 1px 1px #ccc inset;
  -webkit-box-shadow: 0 1px 1px #ccc inset;
  box-shadow: 0 1px 1px #ccc inset;
}

```

```

}

#input input:focus, #input select:focus {
  background-color: #fff;
  border-color: #4fa1fc;
  outline: none;
  -moz-box-shadow: 0 0 0 1px #4fa1fc inset;
  -webkit-box-shadow: 0 0 0 1px #4fa1fc inset;
  box-shadow: 0 0 0 1px #4fa1fc inset;
}

#input label {
  position: relative;
  top: 2px;
  right: 2px;
}

#sub {
  position: relative;
}

.wrap {
  width: 85%;
  margin: auto;
}

.test-data {
  border: 10px solid #f3f3f3;
  padding: 10px;
  margin: 20px;
}

.q {
  border-bottom: 1px solid #000;
}

.a {
  padding: 5px 0;
}

.question {
  display: none;
}

.pagination {
  margin: 20px;
  text-align: center;
}

```

```
.pagination a{
  border: 1px solid #ddd;
  border-radius: 50%;
  padding: 6px 12px;
  text-decoration: none;
  background: #fff;
  color: #428bca;
  margin: 0 5px 0 0;
  display: inline-block;
}
```

```
.pagination a:hover {
  background: #eee;
}
```

```
.pagination > a.nav-visited {
  background-color: #d8d8d5;
}
```

```
.pagination > a.nav-active {
  background: #420bca;
  color: #fff;
  border-color: #428bca;
  cursor: default; }
.center {
  display: block;
  margin: auto;
  text-align: center;
}
```

```
button.btn{
  border: 1px solid #ccc;
  border-radius: 5px;
  cursor: pointer;
  color: white;
  text-decoration: none;
  height: 30px;
  padding: .5em 2em;
  border-width: 2px 0;
  border-color: #4fa1fc #000 #4fa1fc;
  background: linear-gradient(#4fa1fc, #4fa1fc) #4fa1fc;
  transition: 0.2s;
  width: 238px;
  margin: 20px auto;
}
```

```
button.btn:hover {
  background: #a2ceff linear-gradient(#a2ceff, #6ca9ec);
}
```

```

    box-shadow: 0 0 2px #aaa;
}

.none{
    display: none;
}

.a-correct_answer {
    display: none;
}

.a-type_question {
    display: none;
}

.a-parent_question {
    display: none;
}

/*стили таймера*/
#timer_wrap {
    background-color: #f3f3f3;
    border-radius:5px;
    height: 36px;
    width: 80px;
    padding: 5px;
    position:fixed;
    bottom: 70px;
    right: 5%;
}

#stop {
    color: #4fa1fc;
    font-size: 2em;
    position:fixed;
}

.kmenu {
    width: 20%;
    float: left;
}

.kmenu ul {
    list-style: none;
}

.kcontent {
    width: 80%;
    float: right;
}

```

```

}

table {
    border-spacing: 7px 11px;
    margin: 5px;
    border-collapse: collapse;
}
td {
    padding: 5px; /* Поля вокруг текста */
    border: 1px solid black;
}

.sprafka {
    margin: 15px 55px;
    line-height: 140%;
    text-align: justify;
}

.sprafka ul li {
    margin: 5px 0 5px 40px;
}

h2 {
    margin-left: 30px;
}
h3 {
    margin-left: 30px;
    color: black;
}

```

## **function.php**

```

<?php
function get_facility(){
    global $db;
    $query = "SELECT * FROM medical_facility";
    $res = mysqli_query($db, $query);
    if (!$res) return false;
    $data = array();
    while ($row = mysqli_fetch_assoc($res)) {
        $data[] = $row;
    }
    return $data;
}

function get_category(){
    global $db;
    $query = "SELECT * FROM categories";
    $res = mysqli_query($db, $query);
}

```



```

        if (!$res) return false;
        $data = array();
        while ($row = mysqli_fetch_assoc($res)) {
            $data[] = $row;
        }
        return $data;
    }

function getSpecial(){
    global $db;
    $code = mysqli_real_escape_string($db, $_POST['code']);
    $query = "SELECT *
                FROM specialties
                WHERE parent_group = '$code'";
    $res = mysqli_query($db, $query);
    $data = '';
    while($row = mysqli_fetch_assoc($res)){
        $data .= "<option value='{ $row['id']}'>{ $row['specialty']}</option>";
    }
    return $data;
    return $row;
}

function get_spec(){
    global $db;
    $query = "SELECT * FROM specialties";
    $res = mysqli_query($db, $query);
    if (!$res) return false;
    $data = array();
    while($row = mysqli_fetch_assoc($res)){
        $data[] = $row;
    }
    return $data;
}

function get_group(){
    global $db;
    $query = "SELECT * FROM groups";
    $res = mysqli_query($db, $query);
    if (!$res) return false;
    $data = array();
    while ($row = mysqli_fetch_assoc($res)){
        $data[] = $row;
    }
    return $data;
}

function get_test_data($id_spec){
    if (!$id_spec) return;
    global $db;

```

```

$sql = "SELECT *
        FROM tests
        ORDER BY test_id DESC";
$result = mysqli_query($db, $sql);
$d = mysqli_fetch_assoc($result);
$_SESSION['test_id'] = $d['test_id'];

$query = "SELECT q.question, q.specialty, a.id, a.answer, a.parent_question,
a.correct_answer, q.type_question
        FROM questions q
        LEFT JOIN answers a
        ON q.id = a.parent_question
        WHERE q.specialty = $id_spec";
$res = mysqli_query($db, $query);
$data1 = NULL;
$data = NULL;
while ($row = mysqli_fetch_assoc($res)) {
    $data1[$row['parent_question']]['parent_question'] = $row['parent_question'];
    $data1[$row['parent_question']][0] = $row['question'];
    $data1[$row['parent_question']][$row['id']] = $row['answer'];
    if ($row['correct_answer'] == 1)
        {$data1[$row['parent_question']]['correct_answer'] = $row['id'];}
    $data1[$row['parent_question']]['type_question'] = $row['type_question'];
}

shuffle($data1);
global $count_quest;
if ($_SESSION['type_test'] == 'probe') {
    $count_quest = 15;
}
for ($i=0; $i < $count_quest; $i++) {
    $data[$i] = $data1[$i];
}
return $data;
}

function get_all_test_data($id_spec){
    if (!$id_spec) return;
    global $db;
    $query = "SELECT q.question, q.specialty, a.id, a.answer, a.parent_question,
a.correct_answer
            FROM questions q
            LEFT JOIN answers a
            ON q.id = a.parent_question
            WHERE q.specialty = $id_spec";
    $res = mysqli_query($db, $query);
    $data = NULL;
    while ($row = mysqli_fetch_assoc($res)) {
        $data[$row['parent_question']]['parent_question'] = $row['parent_question'];
    }
}

```

```

        $data[$row['parent_question']][0] = $row['question'];
        $data[$row['parent_question']][$row['id']] = $row['answer'];
        if ($row['correct_answer'] == 1)
            {$data[$row['parent_question']]['correct_answer'] = $row['id'];}
    }
    return $data;
}

```

```

function get_test_result($test_data, $result, $POST){
    unset($POST['spec']);
    if(isset($test_data)){
        foreach($POST as $q => $a){
            foreach ($test_data as $test) {
                if ($a['parent_quest'] == $test['parent_question']) {
                    if (isset($a['user_answer'])) {
                        if ($a['user_answer'] != $test['correct_answer']){
                            $POST[$q]['incorrect_answer'] = $a['user_answer'];
                        }
                        else {
                            $POST[$q]['correct_answer'] = $a['user_answer'];
                        }
                    }
                    else {
                        $POST[$q]['incorrect_answer'] = -1;
                    }
                }
            }
        }
    }
    return $POST;
}

```

```

function print_result($test_result, $test_id, $user_id){

    $all_count = count($test_result);//количество вопросов в тесте
    $correct_answer_count = 0;//количество верных ответов
    $incorrect_answer_count = 0;//количество неверных ответов
    $unanswer_count = 0; //количество не отвеченных вопросов
    $percent = 0;//процент верных ответов
    if(isset($test_result)){
        foreach ($test_result as $item) {
            if (isset($item['correct_answer'])) $correct_answer_count++;
            elseif ($item['incorrect_answer'] != -1) {
                $incorrect_answer_count++;
            }
        }
        $unanswer_count = $all_count - $correct_answer_count - $incorrect_answer_count;
        $percent = round(($correct_answer_count/$all_count*100),2);
        $print_res = '<div class="count-res">';
        $print_res .= '<div class="test-data">';
    }
}

```

```

$print_res .= "<p>Всего вопросов: <b>{$all_count}</b></p>";
$print_res .= "<p>Отвечено верно: <b>{$correct_answer_count}</b></p>";
$print_res .= "<p>Отвечено не верно: <b>{$incorrect_answer_count}</b></p>";
$print_res .= "<p>Осталось не отвеченными: <b>{$unanswer_count}</b></p>";
$print_res .= "<p>Верных ответов: <b>{$percent} %</b></p>";
if ($percent>=70) $print_res .= "<p><b>ТЕСТИРОВАНИЕ ПРОЙДЕНО!</b></p>";
else $print_res .= "<p><b>ТЕСТИРОВАНИЕ НЕ ПРОЙДЕНО!</b></p>";
$print_res .= '</div>';
$print_res .= '</div>';
$print_res .= '<a href="kabinet.php"><button class="center btn"
id="exit">Закрыть</button></a>';

```

```

global $db;
date_default_timezone_set('Asia/Krasnoyarsk');
$date = date("Y-m-d H:i");
$result = $correct_answer_count;
$result .= "/";
$result .= $all_count;
$type_test = $_SESSION['type_test'];
$specialty = $_SESSION['id_spec'];
$category = $_SESSION['id_category'];
$query = "INSERT INTO test_results (date, user_id, test_id, result, percent, type_test,
specialty_t, category_t)
VALUES ('$date', '$user_id', '$test_id', '$result', '$percent', '$type_test', '$specialty', '$category')";
mysqli_query($db, $query);
return $print_res;
}}

```

```

function insert_result($test_result, $type_test, $test_id, $user_id){
global $db;
foreach ($test_result as $k => $v) {
    if (isset($v['incorrect_answer'])) {
        $user_answer = $v['incorrect_answer'];
    }
    else $user_answer = $v['user_answer'];
    $question_id = $v['parent_quest'];
    $query = "INSERT INTO tests (test_id, question_id, user_answer)
VALUES ('$test_id', '$question_id', '$user_answer')";
    mysqli_query($db, $query);
    if ($type_test == 'probe') {
        $sql = "UPDATE users
SET probe_test = '1'
WHERE user_id = '$user_id' ";
        mysqli_query($db, $sql);
    }
    if ($type_test == 'main') {
        $sql = "UPDATE users
SET main_test = '1'
WHERE user_id = '$user_id' ";
    }
}
}

```

```

        mysqli_query($db, $sql);
    }
    if ($type_test == 'additional') {
        $sql = "UPDATE users
                SET main_test1 = '1'
                WHERE user_id = '$user_id' ";
        mysqli_query($db, $sql);
    }
    $sql = "SELECT probe_test, main_test, main_test1
            FROM users
            WHERE user_id = '$user_id' ";
    $result = mysqli_query($db, $sql);
    $data = mysqli_fetch_assoc($result);
    $_SESSION['probe_test'] = $data['probe_test'];
    $_SESSION['main_test'] = $data['main_test'];
    $_SESSION['main_test1'] = $data['main_test1'];
}
}

function status ($user_id){
    global $db;
    $sql = "SELECT probe_test, main_test, main_test1
            FROM users
            WHERE user_id = '$user_id' ";
    $result = mysqli_query($db, $sql);
    $data = mysqli_fetch_assoc($result);
    $_SESSION['probe_test'] = $data['probe_test'];
    $_SESSION['main_test'] = $data['main_test'];
    $_SESSION['main_test1'] = $data['main_test1'];
}

function pagination ($count_questions, $test_data){
    $keys = array_keys($test_data);
    $pagination = '<div class="pagination">';
    for ($i = 1; $i <= $count_questions; $i++){
        $key = array_shift($keys);
        if ($i == 1){
            $pagination .= '<a class="nav-active" href="#question-' . $key . '">' . $i .
'</a>';
        }
        else{$pagination .= '<a href="#question-' . $key . '">' . $i . '</a>';}
        if (($i % 15) == 0){
            $pagination .= '<br>';
        }
    }
    $pagination .= '</div>';
    return $pagination;
}

```

```

function registration($post) {
    global $db;
    $login = clean_data($post['reg_login']);
    $password = trim($post['reg_password']);
    $conf_password = trim($post['reg_conf_password']);
    $email = clean_data($post['reg_email']);
    $surname = clean_data($post['reg_surname']);
    $name = clean_data($post['reg_name']);
    $patronymic = clean_data($post['reg_patronymic']);
    $group = clean_data($post['reg_group']);
    $specialty = clean_data($post['reg_specialty']);
    $experience = clean_data($post['reg_experience']);
    $mf = clean_data($post['reg_mf']);
    $category = clean_data($post['reg_category']);
    $phone = clean_data($post['reg_phone']);
    $consent = clean_data($post['reg_consent']);
    $msg = '';
    if (!isset($_POST['reg_mf'])) {
        $_SESSION['reg']['login'] = $login;
        $_SESSION['reg']['email'] = $email;
        $_SESSION['reg']['surname'] = $surname;
        $_SESSION['reg']['name'] = $name;
        $_SESSION['reg']['patronymic'] = $patronymic;
        $_SESSION['reg']['group'] = $group;
        $_SESSION['reg']['specialty'] = $specialty;
        $_SESSION['reg']['experience'] = $experience;
        $_SESSION['reg']['category'] = $category;
        $_SESSION['reg']['phone'] = $phone;
        $_SESSION['reg']['consent'] = $consent;
        return "Выберите Место работы!";
    }
    if (!isset($_POST['reg_category'])) {
        $_SESSION['reg']['login'] = $login;
        $_SESSION['reg']['email'] = $email;
        $_SESSION['reg']['surname'] = $surname;
        $_SESSION['reg']['name'] = $name;
        $_SESSION['reg']['patronymic'] = $patronymic;
        $_SESSION['reg']['group'] = $group;
        $_SESSION['reg']['specialty'] = $specialty;
        $_SESSION['reg']['experience'] = $experience;
        $_SESSION['reg']['mf'] = $mf;
        $_SESSION['reg']['phone'] = $phone;
        $_SESSION['reg']['consent'] = $consent;
        return "Выберите Категорию!";
    }
    if (!isset($_POST['reg_specialty'])) {
        $_SESSION['reg']['login'] = $login;
        $_SESSION['reg']['email'] = $email;
        $_SESSION['reg']['surname'] = $surname;

```

```

$_SESSION['reg']['name'] = $name;
$_SESSION['reg']['patronymic'] = $patronymic;
$_SESSION['reg']['group'] = $group;
$_SESSION['reg']['experience'] = $experience;
$_SESSION['reg']['mf'] = $mf;
$_SESSION['reg']['category'] = $category;
$_SESSION['reg']['phone'] = $phone;
$_SESSION['reg']['consent'] = $consent;
return "Выберите Специальность!";
}
if($msg) {
    $_SESSION['reg']['login'] = $login;
    $_SESSION['reg']['email'] = $email;
    $_SESSION['reg']['surname'] = $surname;
    $_SESSION['reg']['name'] = $name;
    $_SESSION['reg']['patronymic'] = $patronymic;
    $_SESSION['reg']['group'] = $group;
    $_SESSION['reg']['specialty'] = $specialty;
    $_SESSION['reg']['experience'] = $experience;
    $_SESSION['reg']['mf'] = $mf;
    $_SESSION['reg']['category'] = $category;
    $_SESSION['reg']['phone'] = $phone;
    $_SESSION['reg']['consent'] = $consent;
    return $msg;
}
if ($conf_password == $password) {
    $sql = "SELECT user_id, login
            FROM users
            WHERE login = '$login'";
    $result = mysqli_query($db, $sql);
    if (mysqli_num_rows($result) > 0) {
        $_SESSION['reg']['email'] = $email;
        $_SESSION['reg']['surname'] = $surname;
        $_SESSION['reg']['name'] = $name;
        $_SESSION['reg']['patronymic'] = $patronymic;
        $_SESSION['reg']['group'] = $group;
        $_SESSION['reg']['specialty'] = $specialty;
        $_SESSION['reg']['experience'] = $experience;
        $_SESSION['reg']['mf'] = $mf;
        $_SESSION['reg']['category'] = $category;
        $_SESSION['reg']['phone'] = $phone;
        $_SESSION['reg']['consent'] = $consent;
        return "Пользователь с таким логином уже существует!";
    }
    $sql = "SELECT user_id, email
            FROM users
            WHERE email = '$email'";
    $result = mysqli_query($db, $sql);
    if (mysqli_num_rows($result) > 0) {

```

```

$_SESSION['reg']['login'] = $login;
$_SESSION['reg']['surname'] = $surname;
$_SESSION['reg']['name'] = $name;
$_SESSION['reg']['patronymic'] = $patronymic;
$_SESSION['reg']['group'] = $group;
$_SESSION['reg']['specialty'] = $specialty;
$_SESSION['reg']['experience'] = $experience;
$_SESSION['reg']['mf'] = $mf;
$_SESSION['reg']['category'] = $category;
$_SESSION['reg']['phone'] = $phone;
$_SESSION['reg']['consent'] = $consent;
return "Пользователь с таким e-mail уже существует!";
}
$password = md5($password);
$hash = md5(microtime());
$query = "INSERT INTO users (login, password, hash, email, surname, name,
patronymic, specialty, experience, mf, category, phone, consent)
VALUES ('$login','$password','$hash','$email', '$surname', '$name',
'$patronymic', '$specialty', '$experience', '$mf', '$category', '$phone', '$consent')";
$result2 = mysqli_query($db, $query);
if (!$result2) {
    return "Ошибка при добавлении в базу данных!";
}
else {
    $headers = "";
    $headers .= "From: Admin <admin@mail.ru> \r\n";
    $headers .= "Content-Type: text/plain; charset=utf8> \r\n";
    $tema = "registration";
    $mail_body = "Спасибо за регистрацию. Ваша ссылка:
http://localhost/testing/confirm.php?hash=".$hash;
    mail($email, $tema, $mail_body, $headers);
    return TRUE;
}
}
else {
    $_SESSION['reg']['login'] = $login;
    $_SESSION['reg']['surname'] = $surname;
    $_SESSION['reg']['name'] = $name;
    $_SESSION['reg']['patronymic'] = $patronymic;
    $_SESSION['reg']['group'] = $group;
    $_SESSION['reg']['specialty'] = $specialty;
    $_SESSION['reg']['experience'] = $experience;
    $_SESSION['reg']['mf'] = $mf;
    $_SESSION['reg']['category'] = $category;
    $_SESSION['reg']['phone'] = $phone;
    $_SESSION['reg']['consent'] = $consent;
    $_SESSION['reg']['email'] = $email;
    return "Вы неправильно подтвердили пароль!";
}
}

```



```

}

function clean_data($str) {
    return strip_tags(trim($str));
}

function confirm() {
    global $db;
    $new_hash = clean_data($_GET['hash']);
    $new_hash = mysqli_real_escape_string($db, $new_hash);
    $sql = "SELECT surname, name, patronymic, mf
            FROM users
            WHERE hash = '$new_hash'";
    $res = mysqli_query($db, $sql);
    $data = mysqli_fetch_assoc($res);
    $surname = $data['surname'];
    $name = $data['name'];
    $patronymic = $data['patronymic'];
    $mf = $data['mf'];
    $sql1 = "SELECT id
            FROM expected
            WHERE surname = '$surname'
              AND name = '$name'
              AND patronymic = '$patronymic'
              AND mf = '$mf'";
    $result1 = mysqli_query($db, $sql1);
    if (mysqli_num_rows($result1) == 0) {
        $_SESSION['msg'] = "Такого пользователя нет в списке ожидаемых
        тестируемых";
        return false;
    }
    else {
        $query = "UPDATE users
                  SET confirm = '1', confirm1 = '1'
                  WHERE hash = '$new_hash'";
        $result = mysqli_query($db, $query);
        if (mysqli_affected_rows($db) != 1) {
            $_SESSION['msg'] = "Не верный код подтверждения регистрации!";
            return false;
        }
        else {
            return TRUE;
        }
    }
}

function check_user(){
    global $db;
    if (isset($_SESSION['sess'])) {

```

```

        $sess = $_SESSION['sess'];
        $sql = "SELECT user_id
                FROM users
                WHERE session = '$sess'";
        $result = mysqli_query($db, $sql);
        if (!$result || mysqli_num_rows($result) < 1) {
            return FALSE;
        }
        return TRUE;
    }
    elseif (isset($_COOKIE['login']) && isset($_COOKIE['password'])) {
        $login = $_COOKIE['login'];
        $password = $_COOKIE['password'];
        $sql = "SELECT user_id
                FROM users
                WHERE login = '$login'
                      AND password = '$password'
                      AND confirm = '1'";
        $result2 = mysqli_query($db, $sql);
        if (!$result2 || mysqli_num_rows($result2) < 1) {
            return FALSE;
        }
        $sess = md5(microtime());
        $query = "UPDATE users
                  SET session = '$sess'
                  WHERE login = '$login'";
        if (!mysqli_query($db, $query)) {
            return "Ошибка авторизации пользователя!";
        }
        $_SESSION['sess'] = $sess;
        return TRUE;
    }
    else {
        return FALSE;
    }
}

function login($post){
    global $db;
    $login = clean_data($post['login']);
    $password = md5(trim($post['password']));

    $sql = "SELECT user_id, name, patronymic, confirm, confirm1, probe_test, main_test,
    main_test1, admin, block
            FROM users
            WHERE login = '$login'
                  AND password = '$password'";
    $result = mysqli_query($db, $sql);
    $data = mysqli_fetch_assoc($result);

```

```

if (!$result || mysqli_num_rows($result) < 1) {
    return "Не верный логин или пароль!";
}
if(($data['confirm'] == 0) || ($data['confirm1'] == 0)) {
    return "Пользователь с таким логином еще не подтвержден!";
}
if($data['block'] == 1) {
    return "Пользователь заблокирован! Обратитесь в тех. поддержку!";
}

$sess = md5(microtime());
$query = "UPDATE users
        SET session = '$sess'
        WHERE login = '$login'";

if (!mysqli_query($db, $query)) {
    return "Ошибка авторизации пользователя!";
}
$_SESSION['sess'] = $sess;
$_SESSION['login'] = $login;
$_SESSION['name'] = $data['name'];
$_SESSION['patronymic'] = $data['patronymic'];
$_SESSION['user_id'] = $data['user_id'];
$_SESSION['probe_test'] = $data['probe_test'];
$_SESSION['main_test'] = $data['main_test'];
$_SESSION['main_test1'] = $data['main_test1'];
$_SESSION['a'] = $data['admin'];
if ($post['member'] == 1) {
    $time = time() + 10*24*60*60;
    setcookie('login', $login, $time);
    setcookie('password', $password, $time);
}
return TRUE;
}

function logout() {
    $_SESSION = array();
    setcookie('login', '', time()-3600);
    setcookie('password', '', time()-3600);
    return TRUE;
}

function get_password($email) {
    global $db;
    $email = clean_data($email);
    $sql = "SELECT user_id
            FROM users
            WHERE email = '$email'";

```

```

$result = mysqli_query($db, $sql);
$data = mysqli_fetch_assoc($result);
$user = $data['user_id'];
if (!$user) {
    return "Не возможно восстановить пароль. Пользователя с таким почтовым
ящиком нет!";
}
if(mysqli_num_rows($result) == 1) {
    $str = "234567890qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";
    $pass = "";
    for ($i = 0; $i < 6 ; $i++) {
        $x = mt_rand(0, (strlen($str)-1));
        $pass .= $str[$x];
    }
    $md5pass = md5($pass);
    $query = "UPDATE users
                SET password = '$md5pass'
                WHERE user_id = '$user'";
    $result2 = mysqli_query($db, $query);
    if (!$result2) {
        return "Невозможно сгенерировать новый пароль. Ошибка при обращении к
БД.";
    }
    $headers = "";
    $headers .= "From: Admin <admin@mail.ru>";
    $headers .= "Content-Type:text/plain;charset=utf8";
    $subject = 'new password';
    $mail_body = "Ваш новый пароль: ".$pass;
    mail($email, $subject, $mail_body, $headers);
    return TRUE;
}
}

```

Выпускная квалификационная работа выполнена мной самостоятельно.  
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография \_\_\_\_\_ наименований.

Один экземпляр сдан на кафедру.

«\_\_\_\_\_» \_\_\_\_\_ 2017 г.  
дата

\_\_\_\_\_  
подпись

В. С. Дресвянкин  
инициалы, фамилия